



**Technische
Universität
Braunschweig**



Ein Monitoring-System zur automatischen Dokumentation von ACAS/TCAS-Eingriffen in den Luftverkehr

Jens Gottstein

TU Braunschweig

Institut für Eisenbahnwesen und Verkehrssicherung

CFF - Forschungsbericht 2010-06

TU Braunschweig - Campus Forschungsflughafen

TU Braunschweig - Campus Forschungsflughafen

Berichte aus der Luft- und Raumfahrttechnik

Forschungsbericht 2010-06

**Ein Monitoring-System zur automatischen Dokumentation
von ACAS/TCAS-Eingriffen in den Luftverkehr**

Jens Gottstein

TU Braunschweig

Institut für Eisenbahnwesen und Verkehrssicherung

Diese Arbeit erscheint gleichzeitig als von der Fakultät Architektur, Bauingenieurwesen und Umweltwissenschaften der Technischen Universität Carolo-Wilhelmina zu Braunschweig zur Erlangung des akademischen Grades eines Doktor-Ingenieurs genehmigte Dissertation.

Ein Monitoring-System zur automatischen Dokumentation von ACAS/TCAS-Eingriffen in den Luftverkehr

Von der
Fakultät Architektur, Bauingenieurwesen und Umweltwissenschaften
der Technischen Universität Carolo-Wilhelmina
zu Braunschweig

zur Erlangung des Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
genehmigte

Dissertation

von
Jens Gottstein
geboren am 22. Dezember 1978
aus Pößneck

Eingereicht am 27. Januar 2010
Disputation am 09. September 2010

Berichterstatter Prof. Dr.-Ing. Peter Form
Prof. Dr.-Ing. Wolfgang Niemeier

Von der Fakultät Architektur, Bauingenieurwesen und Umweltwissenschaften
der Technischen Universität Carolo-Wilhelmina zu Braunschweig
zur Erlangung des Grades eines Doktoringenieurs (Dr.-Ing.) genehmigte Dissertation

Eingereicht am 27. Januar 2010
Disputation am 09. September 2010
Berichterstatter Prof. Dr.-Ing. Peter Form
Prof. Dr.-Ing. Wolfgang Niemeier

Institut für Eisenbahnwesen und Verkehrssicherung
Technische Universität Braunschweig
Pockelsstraße 3 • 38106 Braunschweig

<http://www.digibib.tu-bs.de>

Herausgeber der CFF Forschungsberichte:

TU Braunschweig - Campus Forschungsflughafen
Hermann-Blenk-Straße 27 • 38108 Braunschweig
Tel: 0531-391-9822 • Fax: 0531-391-9808
<http://www.campus-forschungsflughafen.de>
<mailto:cff@tu-braunschweig.de>

ISBN 13: 978-3-928628-56-3

Die Deutsche Bibliothek – CIP Einheitsaufnahme

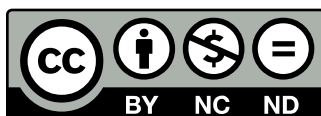
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie;
detaillierte bibliographische Daten sind im Internet über <http://www.d-nb.de> abrufbar.

Dieses Werk wird außerdem im Selbstverlag herausgegeben von:

Jens Gottstein
Kastanienallee 37 • 38104 Braunschweig
<mailto:jens.gottstein@gmx.de>

Umschlagbild: http://commons.wikimedia.org/wiki/File:Air_traffic_heathrow.JPG
von Nutzer „Kierano“ veröffentlicht unter „Creative Commons Attribution-Share Alike 3.0“ Lizenz

Dieses Werk ist unter einem „Creative Commons Namensnennung – Keine kommerzielle Nutzung
– Keine Bearbeitung 3.0 Deutschland“-Lizenzvertrag lizenziert. Die Lizenz kann per Brief an Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA angefordert werden oder unter <http://creativecommons.org/licenses/by-nc-nd/3.0/de/> eingesehen werden.



Lange vor dieser Arbeit habe ich das Ansprechen des Bordkollisionsschutzsystems als Passagier selbst erlebt und als vorausschauende Reaktion auf eine eventuell eintretende Gefahr empfunden. Als Informatiker und aus eigener Erfahrung bin ich von autonomen Pilotenassistenzsystemen fasziniert. Professor Form am Institut für Eisenbahnwesen und Verkehrssicherung der Technischen Universität Braunschweig hat diese Arbeit angeregt und mit persönlichem Engagement unterstützt. Er ermöglichte die Arbeit an realen Kommunikationen der realen Bordkollisionsschutzsysteme. Der andauernde wissenschaftliche Diskurs mit ihm hat wesentlich zum Gelingen dieser Arbeit beigetragen.

Ebenfalls danke ich meinem zweiten Berichterstatter Professor Niemeier, sowie den Mitgliedern der Prüfungskommission Professor Wermuth und Professor Friedrich für die kritische Durchsicht meiner Arbeit. Mein Dank gilt gleichfalls meinen ehemaligen Kollegen am Institut für Eisenbahnwesen und Verkehrssicherung für die Integration in ihr Team, trotz eines anderen Themenschwerpunktes. Ich danke den neuen Kollegen und insbesondere Professor Hecker für die herzliche Aufnahme in das Institut für Flugführung, wo ich meine gesammelten Erfahrungen in verwandte Projekte im Bereich ADS-B und Mode S einbringen kann. Den Studenten Alexander Burkert, Stephanie Pietsch, Dorothee Schröder und Christopher Josephs danke ich für die Mitarbeit am ACAS-Monitoring und verwandten Untersuchungen.

Für die Aufnahme von Monitorstationen danke ich Herrn Klein und Professor Kügler vom DLR-Braunschweig, Professor Günther vom DLR-Oberpfaffenhofen, Frau von Eckartsberg und Herrn Landgraf von FRAPORT, sowie Holger Neufeld von THALES.

Meinen Freunden danke ich für die Übernahme des Lektorats als Crowdsourcing-Aufgabe, besonderer Dank gilt dabei Katrin Gottstein, die den größten Anteil übernahm. Meinen Eltern danke ich für Unterstützung und Zuspruch bei den beruflichen Richtungsentscheidungen, die während des Projektes zu treffen waren. Ganz besonders möchte ich meiner Frau Claudia danken, die mit vielen Ermutigungen, ausdauerndem Lektorat und dem richtigen Maß an Ablenkung den Abschluss dieser Arbeit ermöglicht hat.

Inhaltsverzeichnis

1 Einführung in die Problematik.....	1
1.1 Methodik und Aufbau der Arbeit.....	2
1.2 ACAS-Grundlagen und deren Entwicklung.....	4
2 ACAS-Funktion und Kommunikation sowie die Möglichkeiten der Beobachtung.....	5
2.1 ACAS-Funktion und Kommunikation.....	5
2.2 ACAS-Monitoring-Idee.....	9
2.3 Mode S-Übertragungsprotokoll.....	9
2.4 Ausschluss von Übertragungsfehlern – CRC-Prüfsumme im AP-Feld.....	11
2.5 Aufbau der Mode S-Formate im Überblick.....	14
2.6 Mode S-Formate des Mode S-Radars.....	16
2.7 Codierung der ACAS-Informationen im Mode S-Protokoll.....	16
2.8 Selbsttätige Aussendungen der Flugzeuge – Extended Squitter – ADS-B.....	22
2.9 RA-Abfrage mittels Mode S-Radar – RA-Downlink.....	23
3 Der ACAS-Monitoring-Ansatz.....	25
4 Empfang von Mode S-Nachrichten – Anbindung von Mode S-Empfangsstationen.....	27
4.1 Experimentelle Empfangsanlage der TU Braunschweig.....	27
4.2 Thales Empfangsanlage.....	29
4.3 Empfangsstatistiken.....	34
4.4 Reichweite einer Station.....	38
5 Analyse von Mode S-Nachrichten zum Aufbau eines umfassenden Umgebungsmodells.....	39
5.1 Speichersparende Architektur – Datenstromverarbeitung.....	39
5.2 Umgebungsmodell – Klasse Plane_DB und ihre Helfer.....	40
5.3 Nachrichtenweise Verarbeitung – Hauptschleife.....	43
5.4 Klasse Mode_s_msg – Mode S-Nachricht.....	45
5.5 Analysezeit – Klasse Global_Clock.....	47
5.6 Funktionen interpret_X() und Verwandte.....	49
6 Zusammenstellen von ACAS-Nachrichten zu ACAS-Event-Berichten.....	51
6.1 Speicherung von ACAS-Events – die Klassen ACAS_Events_DB und ACAS-Event.....	51
6.2 ACAS-Nachrichten zu ACAS-Events zusammenstellen.....	52
6.3 Ausgabe von Berichten.....	56
7 Verarbeitung von Höhen und Positionen als Zusatz zur Interpretation von ACAS-Events.....	57
7.1 Trennung von vertikalen und horizontalen Spuren.....	57
7.2 Höhen.....	58
7.3 Vergleich von Mode A/C-Radar, Mode S-Radar und ADS-B-Positionsdaten.....	59
7.4 ADS-B-Decodierung.....	63
7.5 ADS-B-Filterung, Spurinitiierung und Spurbildung.....	65
8 Vernetzung von ACAS-Monitor-Stationen.....	67
8.1 Mögliche Architekturen des ACAS-Monitor-Netzes.....	67
8.2 Standorte der Bodenstationen.....	70
8.3 Reichweite des Netzes.....	81
8.4 Datenübertragung von SPU's zur Zentrale.....	81
8.5 Sicherheitskonzept des ACAS-Monitor-Netzes.....	85
8.6 Automatisierte Auswertung in der Zentrale in Braunschweig.....	86
8.7 Zusammenfassung des vernetzten ACAS-Monitorings.....	88

9 Beobachtete ACAS-Events.....	91
9.1 Aufbau der ACAS-Event-Berichte.....	91
9.2 ACAS-Event Beispiel 1 – Einfacher Konflikt.....	93
9.3 ACAS-Event Beispiel 2 – Abschwächung des RA nach prompter Reaktion.....	99
9.4 ACAS-Event Beispiel 3 – Hohe vertikale Rate.....	102
9.5 ACAS-Event Beispiel 4 – Ungenaue Höhenhaltung.....	104
9.6 ACAS-Event Beispiel 5 – Flugzeuge fliegen übereinander, oberes sinkt zuerst.....	106
9.7 ACAS-Event Beispiel 6 – Startendes Flugzeug in Konflikt mit landendem.....	108
9.8 ACAS-Event Beispiel 7 – Einschwenken auf Doppelbahnsystem.....	110
9.9 ACAS-Event Beispiel 8 – Preventive RA.....	111
9.10 ACAS-Event Beispiel 9 – Altitude Crossing RA.....	112
9.11 ACAS-Event Beispiel 10 – Sense Reversal.....	114
9.12 ACAS-Event Beispiel 11 – Verdrängung aus dem Reiseflug.....	116
9.13 ACAS-Event Beispiel 12 – Langsame Reaktion.....	118
9.14 ACAS-Event Beispiel 13 – Keine Reaktion.....	120
9.15 ACAS-Event Beispiel 14 – Gegenteilige Reaktion.....	122
9.16 ACAS-Event Beispiel 15 – Horizontales Ausweichen, stärkere Reaktion als angewiesen.....	124
9.17 ACAS-Event Beispiel 16 – Überstarke Reaktion.....	126
9.18 ACAS-Event Beispiel 17 – Nicht ACAS 7.0 Ausrüstung.....	128
9.19 ACAS-Event Beispiel 18 – Beabsichtigte Annäherung – Luftbetankung.....	132
9.20 Statistik der automatischen Dauerauswertung.....	133
9.21 Orte der beobachteten ACAS-Events.....	134
10 Zusammenfassung.....	135
Literaturverzeichnis.....	136
Literatur aus der Flugsicherung.....	136
Literatur aus der Informatik.....	139
Eigene Beiträge.....	142
Verwendete Abkürzungen und Einheiten.....	143

1 Einführung in die Problematik

Nahezu alle Verkehrsflugzeuge sind heute mit einem ACAS II¹- bzw. TCAS II²-Kollisionsschutzgerät ausgestattet. Das ACAS erteilt dem Piloten eine Ausweichanweisung (Resolution Advisory, RA) wenn die Gefahr besteht, in den nächsten 15 bis 35 Sekunden mit einem anderen Flugzeug zu kollidieren [ICAO Annex10v4].

1993 verabschiedete die Internationale Zivilluftfahrtorganisation (ICAO) den ersten weltweit gültigen ACAS-Standard [ICAO93]. ACAS erfuhr bis kurz vor der Standardisierung noch grundlegende Änderungen. Daher beschloss die ICAO, ACAS wegen seiner Komplexität auch nach der weltweiten Einführung weiter zu beobachten und gegebenenfalls anzupassen. Die Einführung von ACAS hatte zum Ziel, die Kollisionshäufigkeit um den Faktor zehn zu verringern: Statt einer Kollision pro 100 Millionen Flugstunden darf dies zukünftig nur noch ein Mal pro eine Milliarde Flugstunden geschehen [ICAO98]. Um dies experimentell nachzuweisen, müsste wirklichkeitsnaher Flugverkehr mit ACAS-ausgerüsteten Flugzeugen über Milliarden von Flugstunden nachgestellt werden. Ein solches Experiment war jedoch nicht durchführbar – solche Langzeitbeobachtungen waren überhaupt erst **nach** der Einführung von ACAS denkbar. So musste ACAS ohne vollständigen Funktionsnachweis eingeführt werden. Aus diesen Gründen rief die ICAO mit der Einführung ein paralleles weltweites Erprobungsprogramm für ACAS ins Leben [ICAO91]. Es verlangte, dass Piloten und Controller jedes Ansprechen von ACAS protokollieren und dass, zumindest in den Industrieländern, ACAS-Anweisungen durch besondere mithörende Bodenstationen empfangen, aufgezeichnet und ausgewertet werden. Die Piloten- und Controllerprotokolle wurden zwar erstellt, gesammelt und ausgewertet, Beobachtungsstationen wurden jedoch nicht entwickelt. Im Rahmen dieser Arbeit werden erstmalig Bodenstationen entwickelt, die, wie von der ICAO gefordert, ACAS-Signale empfangen und extrahieren. Über die ICAO Forderungen hinaus werden die Bodenstation den kompletten ACAS-Koordinations-Dialog und die Verkehrssituation beim ACAS-Eingriff rekonstruieren.

ACAS wurde in den 1990er Jahren weltweit eingeführt. Erkenntnisse aus den Piloten- und Controllerprotokollen flossen in die ACAS-Standards zurück. 1997 wurde ACAS Version 7.0 eingeführt.

Am 01. Juli 2002 kollidierten zwei mit ACAS ausgerüstete Flugzeuge nahe Überlingen, Deutschland [BFU04]. Dieser Unfall und einige weniger folgenschwere Zwischenfälle zeigen deutlich, dass die Weiterentwicklung von ACAS noch nicht abgeschlossen ist, und insbesondere, dass das Eingreifen von ACAS in den von der Flugsicherung kontrollierten Luftraum jedes Mal

1 ACAS II bzw. TCAS II sind die beiden Bezeichnungen des weltweit verpflichtend eingeführten Bordkollisionsschutzsystems. ACAS I bzw. TCAS I wurde nie verpflichtend, ACAS III wurde nie entwickelt. Wenn in der Luftfahrt heute von ACAS bzw. TCAS gesprochen wird, ist ACAS II bzw. TCAS II gemeint, diese Arbeit macht da keine Ausnahme. Im Folgenden bezeichnet der Term ACAS immer ACAS II.

2 ACAS ist die Bezeichnung des internationalen und europäischen Standards für Bordkollisionsschutzgeräte. TCAS ist die Bezeichnung des nationalen Standards in der USA. Alle Hersteller von Geräten nach dem ACAS-Standard vertreiben ihre Produkte unter der Bezeichnung TCAS. Da sich diese Arbeit an dem internationalen Standard orientiert, wird im Folgenden nur noch von ACAS gesprochen.

dokumentiert werden muss. In der Folge wurde das SATT³-Projekt gestartet, in dem die von der ICAO geforderten Bodenstationen entwickelt und erprobt wurden.

Im SATT-Projekt wurde noch ein weiterer Gesichtspunkt deutlich: Das Bordkollisionsschutzsystem ACAS soll ein RA erst in dem Moment anweisen, in dem mit standardisierten Beschleunigungen gerade noch ausgewichen werden kann. Diese Vorwarnzeit führt dazu, dass ACAS gelegentlich vor dem Erreichen der gesetzlichen Staffelung anspricht, und so Flugzeuge zu Manövern zwingt, die den Anweisungen der Flugsicherung nicht entsprechen. Diese Arbeit soll dabei helfen, diese Inkompatibilität präzise messbar zu machen.

Daher wird in dieser Arbeit erforscht, wie mit vertretbarem technischen Aufwand die Eingriffe der existierenden ACAS-Geräte in den realen Luftverkehr automatisch, objektiv und umfassend dokumentiert werden können.

Zur Überwachung der Umgebung fragt ein ACAS-Gerät im Flugzeug ständig alle Flugzeuge in der Umgebung nach ihren Höhen ab, aus den Signallaufzeiten werden die Abstände berechnet. Stellt ein ACAS-Gerät anhand der Höhendifferenz, Änderung der Höhendifferenz, Abstand und Relativgeschwindigkeit eine Kollisionsdrohung fest, so wird ein Resolution Advisory (Ausweichanweisung, RA) an den Piloten gegeben. Ist das andere Flugzeug auch mit ACAS ausgerüstet, dann werden die Ausweichmanöver der Flugzeuge untereinander durch automatische Koordinationsnachrichten (Datenpakete) abgestimmt. Daneben senden die ACAS-Geräte kurze Statusmeldungen an etwaig vorhandene Bodenstationen aus.

In dieser Arbeit werden die Algorithmen und die Software für ein ACAS-Monitoring-System entwickelt, das all diese automatischen Aussendungen mithört und den Ablauf des ACAS-Eingriffes rekonstruiert. Das umfasst die für Bodenstationen vorgesehenen Statusmeldungen, eine Rekonstruktion des Koordinationsdialoges und eine Beschreibung der Verkehrssituation, mit sekundengenauen Höhenverläufen und Bahnen der Flugzeuge. So wird die Verkehrssituation vor dem ACAS-Eingriff, das Resolution Advisory selbst und die Reaktion der Flugzeuge detailliert dokumentiert.

1.1 Methodik und Aufbau der Arbeit

Im Mittelpunkt dieser Arbeit steht die Entwicklung eines ACAS-Monitoring-Systems das Mode S⁴-Signale mithört und daraus Berichte zu ACAS-Eingriffen (sog. ACAS-Event-Berichte) erzeugt.

Bei ACAS-Eingriffen handelt es sich um seltene Ausnahmeereignisse. Um dennoch eine statistisch relevante Anzahl von Vorfällen zu erhalten, müssen große Lufträume über lange Zeiträume beobachtet werden. Die Analyse ist wegen der anfallenden Datenmengen (ca. 1 GByte pro Tag und Empfangsstation) nicht manuell möglich, sondern das ACAS-Monitoring muss vollständig automatisiert ablaufen. Die Reichweite einer Station ist durch die Krümmung der Erde begrenzt. Zur Beobachtung großer Lufträume benötigt man daher mehrere Stationen, die in geeigneter Weise verbunden werden müssen.

³ Das SATT-Projekt (Safety Through Transparency) war ein Teilprojekt des Luftfahrtforschungsprogramms III des Bundesministeriums für Wirtschaft und Technologie. Partner waren die Deutsche Flugsicherung DFS, Thales GmbH Stuttgart und das Institut für Eisenbahnwesen und Verkehrssicherung der TU Braunschweig

⁴ Beachte die Schreibweise „Mode S“ mit Leerzeichen, die auch in Zusammensetzungen wie „Mode S-Transponder“ beibehalten wird.

Darum hat diese Arbeit einen interdisziplinären Charakter, es werden Techniken der Informatik auf ein Problem der elektronischen Verkehrssicherung angewendet. Dementsprechend werden Termini der Informatik und der elektronischen Verkehrssicherung gemeinsam gebraucht. Um die Arbeit verständlich zu halten, werden Grundkenntnisse in beiden Disziplinen vorausgesetzt, alle tiefer gehenden Termini werden für den jeweils fachfremden Leser im Text oder in Fußnoten erklärt.

In Kapitel 2 – „ACAS-Funktion und Kommunikation sowie die Möglichkeiten der Beobachtung“ (Seite 5) werden die nötigen Hintergründe im Mode S- und ACAS-Standard erklärt und daraus Ansätze zum Monitoring abgeleitet. Um einen ACAS-Monitor zu entwickeln, muss geklärt werden, welche Aussendungen Informationen zu ACAS-Eingriffen beitragen können. Wie diese Aussendungen empfangen, validiert und Flugzeugen und Vorfällen zugeordnet werden können und wie die Informationen zu einem umfassenden Bericht zusammengebracht werden können, wird in diesem Kapitel hergeleitet.

Kapitel 3 – „Der ACAS-Monitoring-Ansatz“ (Seite 25) stellt den in Kapitel 2 entwickelten ACAS-Monitoring-Ansatz kompakt dar und skizziert die Umsetzung dieses Ansatzes in der Arbeit.

Kapitel 4 – „Empfang von Mode S-Nachrichten – Anbindung von Mode S-Empfangsstationen“ (Seite 27) beschreibt die im Projekt verwendeten Mode S-Empfangsanlagen und die dazugehörigen Datenformate. Diese Mode S-Empfangsanlagen für 1030 MHz und 1090 MHz liefern die grundlegenden Daten für das ACAS-Monitoring und jeder ACAS-Monitoring-Algorithmus muss an die Gegebenheiten auf den Mode S-Kanälen angepasst sein.

Kapitel 5 – „Analyse von Mode S-Nachrichten zum Aufbau eines umfassenden Umgebungsmodells“ (Seite 39) zeigt, wie aus den empfangenen Mode S-Nachrichten ein mithaltendes Modell des beobachteten Luftraums im Speicher des ACAS-Monitors aufgebaut wird. Dieses Modell wird dann dem eigentlichen ACAS-Monitoring die wichtigen Hintergrundinformationen zu Flugzeugen liefern.

In Kapitel 6 – „Zusammenstellen von ACAS-Nachrichten zu ACAS-Event-Berichten“ (Seite 51) wird ein Algorithmus entwickelt, der die empfangene ACAS-Kommunikation einzelnen Vorfällen zuordnet und die komplette Abfolge der ACAS-Koordination und den Inhalt der ACAS-Anweisungen rekonstruiert. Diese rekonstruierten ACAS-Dialoge werden als ACAS-Events bezeichnet.

Kapitel 7 – „Verarbeitung von Höhen und Positionen als Zusatz zur Interpretation von ACAS-Events“ (Seite 57): Zur vollständigen Beschreibung eines ACAS-Events müssen nicht nur die Anweisungen des ACAS beschrieben sein, sondern auch die Verkehrssituation und die Reaktion der Flugzeuge. Kapitel 7 erklärt, wie diese Informationen gesammelt und den Event-Berichten hinzugefügt werden.

Kapitel 8 – „Vernetzung von ACAS-Monitor-Stationen“ (Seite 67): Die bisherigen Kapitel gingen von einer Empfangsanlage als Datenquelle für das ACAS-Monitoring aus. Aufgrund der Erdkrümmung ist die Reichweite einer Empfangsanlage aber begrenzt und es müssen mehrere Anlagen aufgestellt werden, um ein größeres Gebiet zu beobachten. Dieses Kapitel analysiert mögliche Architekturen eines solchen verteilten ACAS-Monitors und entwickelt einen Ansatz bis zum fertigen, deutschlandweiten ACAS-Monitor-Netz.

Kapitel 9 – „Beobachtete ACAS-Events“ (Seite 91) erklärt den Aufbau der ACAS-Event-Berichte und gibt einen Überblick über die Beobachtungen. Mit den gezeigten Events soll die

Leistungsfähigkeit des ACAS-Monitorings belegt werden. Dazu wird seit dem 20. April 2007 eine kontinuierliche Dauerauswertung von Mode S-Daten durchgeführt, zu Beginn mit einer Station auf dem Institutsdach in Braunschweig, seit August 2008 mit allen fünf Stationen des ACAS-Monitor-Netzes.

Kapitel 10 – „Zusammenfassung“ (Seite 135) schließt die Arbeit ab.

1.2 ACAS-Grundlagen und deren Entwicklung

Bereits in den fünfziger Jahren des 20. Jahrhunderts begann die Air Transport Association of America (ATA) mit der Entwicklung eines Kollisionsschutzsystems. Da bodengestützte Systeme nicht weltweit einführbar und bordseitige Primärradarlösungen zu teuer waren, konzentrierte sich die Entwicklung auf kooperative, rein bordgestützte Systeme [ATA67]. Kooperative Systeme benötigen zur Überwachung der Umgebung zumindest einen Antwort-Transponder in jedem Flugzeug. Kollisionsschutzsysteme würden also erst zuverlässig vor Kollisionen schützen, wenn alle Flugzeuge zumindest mit Antwort-Transpondern ausgerüstet wären. Dieses Einführungsproblem wurde erst in den achtziger Jahren gelöst, als die inzwischen verbreiteten Sekundärradar-Transponder (Mode A/C) in Kollisionsschutzsysteme einbezogen wurden.

Um eine Bedrohung nicht nur zu erkennen, sondern auch ein richtiges Ausweichen zu garantieren, müssen die Ausweichmanöver zweier Flugzeuge koordiniert werden, d. h. auf keinen Fall dürfen beide Flugzeuge in die gleiche Richtung ausweichen. Um den dazu nötigen Luft-Luft-Data-Link zu schaffen, wurde die Standardisierung des neuen Sekundärradarprotokolls Mode S zusammen mit der ACAS-Entwicklung forciert. Mit Mode S wird jedem Flugzeug eine eindeutige Adresse zugewiesen und jedes Flugzeug meldet sich regelmäßig, unabhängig vom Radar, mit dieser Adresse. Damit erfahren ACAS-Systeme die Adressen aller anderen, in der Umgebung befindlichen Flugzeuge und können diese effektiver als mit Mode A/C beobachten und überhaupt erst koordinieren.

Mit Mode S wurde ein Bordkollisionsschutzsystem möglich, das die eigene Umgebung überwacht, eine drohende Kollision rechtzeitig erkennt und eine entsprechende Ausweichanweisung (Resolution Advisory, RA) berechnet, koordiniert und anzeigt. Dabei funktionieren die Kollisionsschutzsysteme ohne jede Ausrüstung am Boden und weltweit in allen Lufträumen. Bereits 1983 standardisierte die RTCA erstmalig ein solches auf Mode S basierendes Kollisionsschutzsystem unter dem Namen TCAS [RTCA DO185]. Die Internationale Zivilluftfahrtorganisation (ICAO) entwickelte aus dem TCAS-Standard den weltweit gültigen ACAS-Standard [ICAO Annex10v4].

Das Institut für Eisenbahnwesen (IfEV, ehemals Institut für Verkehr) begleitete die Entwicklung der ACAS/TCAS-Kollisionsschutzsysteme mit zahlreichen Forschungsarbeiten, die wichtigsten sind die Habilitationsschrift [Form74], die Studie für den Bundesminister für Verkehr [Form83] und die Dissertationen [Schroer84], [Plumeyer86], [Herber91] und [Kügler95].

2 ACAS-Funktion und Kommunikation sowie die Möglichkeiten der Beobachtung

ACAS ist das weltweit standardisierte Kollisionsschutzsystem an Bord von Verkehrsflugzeugen. Es überwacht alle mit Transpondern ausgerüsteten Flugzeuge in seiner Umgebung. Erkennt ACAS eine unmittelbar drohende Kollision, berechnet es ein Resolution Advisory (RA, Ausweichenweisung), koordiniert das RA mit dem bedrohenden Flugzeug und zeigt das RA im Cockpit optisch und akustisch an.

Dieses Kapitel erklärt das Kollisionsschutzsystem ACAS und erarbeitet Möglichkeiten, die Eingriffe des ACAS in den Luftverkehr systematisch zu beobachten und detailliert zu dokumentieren.

2.1 ACAS-Funktion und Kommunikation

Die Überwachung der Umgebung und die Koordination wird durch den Austausch ACAS-spezifischer Nachrichten realisiert, diese sind in Bild 1 dargestellt. ACAS benutzt die Frequenzen (1030 MHz und 1090 MHz), Protokolle und den flugzeugseitigen Antworttransponder des Mode S⁵-Sekundärradars. Zur Vollständigkeit sind in Bild 1 die Bezeichnungen der ACAS-Nachrichten und die Mode S-Format-Nummern⁶ (Nachrichtentypen) angegeben, da das Mode S-Übertragungsprotokoll in der Arbeit von zentraler Bedeutung sein wird. Im Folgenden wird diese ACAS-Kommunikation genauer erklärt. Dabei wird deutlich, welche Kommunikationen mitgehört werden müssen, um das Bordkollisionsschutzsystem zu beobachten.

Jedes Flugzeug besitzt eine weltweit eindeutige 24-Bit-Mode S-Adresse. ACAS nutzt diese Mode S-Adresse, um andere Flugzeuge zu überwachen und anzusprechen. Darum wird beim ACAS-Monitoring jedes Flugzeug anhand seiner Mode S-Adresse identifiziert.

Akquirieren von Mode S-Adressen – DF11-Squitter⁷

Damit ACAS andere Flugzeuge überwachen kann, muss ACAS zunächst die Mode S-Adressen der Flugzeuge in der Umgebung akquirieren. Deshalb sendet jedes Flugzeug alle 0,8 s bis 1,2 s einen sogenannten DF11-Squitter (1 in Bild 1) aus. Dieser enthält die Adresse des Absenders. ACAS hört die DF11-Squitter mit und baut daraus eine Liste der im Luftraum befindlichen Flugzeuge auf. Flugzeuge, die sich eine gewisse Zeit nicht gemeldet haben, werden aus der Liste entfernt. Auch mithörende Bodenstationen können nach diesem Prinzip Mode S-Adressen akquirieren.

5 Beachte die Schreibweise „Mode S“ mit Leerzeichen, die auch in Zusammensetzungen wie „Mode S-Transponder“ beibehalten wird.

6 Das Mode S-Protokoll unterscheidet verschiedene Nachrichten-Formate (Bei anderen Netzwerkprotokollen auch oft als Nachrichtentypen bezeichnet). Es gibt die Uplink-Formate UF00, UF04, UF11, UF16, UF20, UF21 und UF24 im 1030 MHz Kanal, die Downlink-Formate DF00, DF04, DF05, DF11, DF17 bis DF21 und DF24 im 1090 MHz Kanal. Zusätzlich sind noch diverse Unterformate wie UF16-30 und UF16-31 von Bedeutung.

7 Die Schreibweise DF11-Squitter bezeichnet die Mode S-Format-Nummer und den Namen der Nachricht, die beiden Angaben sind gleichbedeutend. Diese redundante Schreibweise wird verwendet, um die Lesbarkeit zu erhöhen. Die Schreibweise UF16-30-Resolution-Message benennt Format-Unterformat-Name.

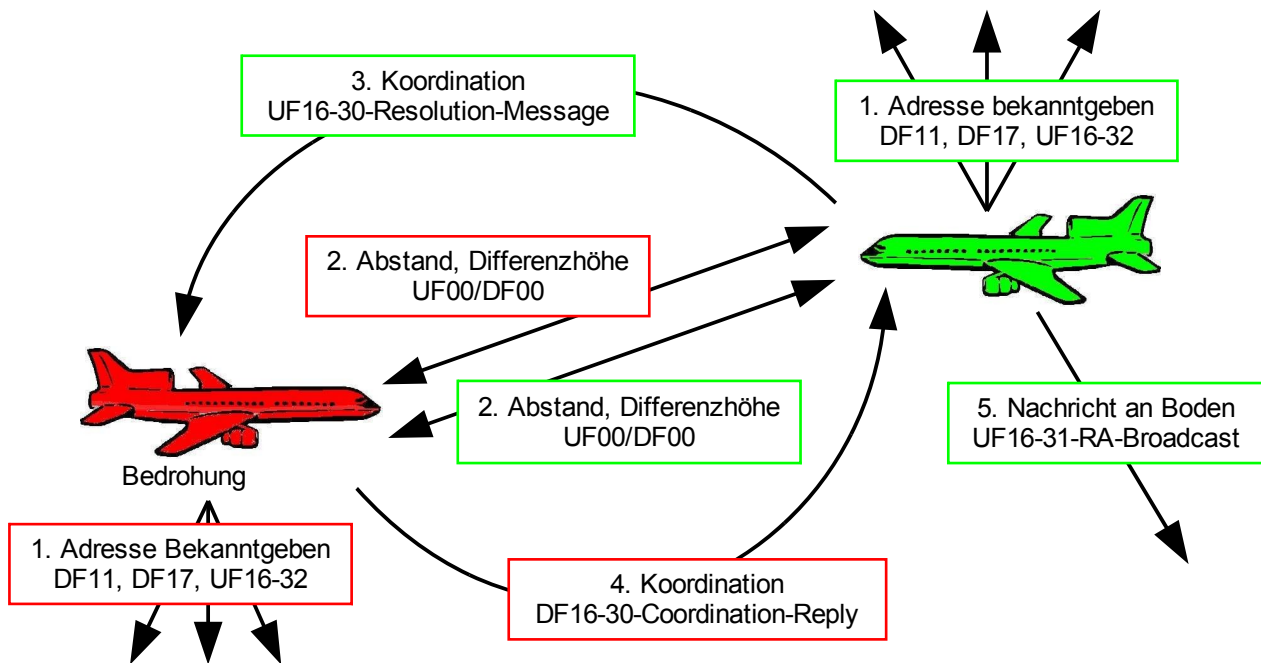


Bild 1: ACAS-Kommunikation in dem Fall, dass das ACAS in Grün ein RA an seinen Piloten anweist, das ACAS in Rot weist (noch) kein RA an

Überwachung der Umgebung – DF00-Air-Air-Surveillance-Antwort

ACAS fragt alle akquirierten Mode S-Adressen mit UF00-Air-Air-Surveillance-Abfragen regelmäßig ab (2 in Bild 1). Die UF00 wird mit einer DF00-Air-Air-Surveillance-Antwort beantwortet, diese enthält die Druckhöhe h und die Adresse des Abgefragten. Aus der Laufzeit des Abfrage-Antwort-Paares berechnet ACAS die Schrägentfernung r . Durch wiederholtes Abfragen bestimmt ACAS die zeitlichen Ableitungen der Relativgeschwindigkeit \dot{r} bzw. der Steig-/Sinkrate $\Delta \dot{h}$ des anderen.

ACAS bestimmt aus den DF00-Air-Air-Surveillance-Antworten auch eine grobe relative Peilung Θ mit 10° Standardabweichung. Diese Genauigkeit ist jedoch nicht ausreichend, um die Peilung in die RA-Berechnung einzubeziehen. So kann ACAS nur vertikale RAs und keine horizontalen Ausweichmanöver berechnen. Die Peilgenauigkeit genügt aber, um dem Piloten die Richtungen zu anderen Flugzeugen im Traffic Display⁸ anzuzeigen.

Ist ein Flugzeug in einem ganz anderen Höhenband als das eigene, so fragt ACAS die Höhe nicht selbst ab, sondern hört die DF00-Air-Air-Surveillance-Antworten auf Abfragen Dritter mit. Dadurch wird weniger Funkfeldlast erzeugt. Auch der ACAS-Monitor kann die DF00-Air-Air-Surveillance-Antworten mithören und damit ein sehr detailliertes Höhenprofil erstellen, viel detaillierter als es einem Radar möglich ist (siehe Abschnitt 7.3). Das in dieser Arbeit entwickelte ACAS-Monitoring kann die Laufzeit des Frage-Antwort-Paares jedoch **nicht** nachvollziehen und kann den Abstand so nicht „mithören“.

⁸ ACAS zeigt dem Piloten alle Flugzeuge der Umgebung in Draufsicht an, dies wird als Traffic Display bezeichnet. Das Traffic Display ist ein eigenes Instrument im Cockpit oder im Hintergrund des Variometers. Im Glas-Cockpit wird das Traffic Display im Navigational Display oder Wetter-Radar-Bild eingeblendet.

Situation-Awareness Erhöhen, Sichtentdeckung unterstützen – Traffic Advisory TA

Um den Piloten bereits vor dem Eintreten eines Resolution Advisory (RA, Ausweichenanweisung) auf das Entstehen einer möglichen Gefährdung hinzuweisen, gibt ACAS ein Traffic Advisory (TA, Verkehrswarnung) an den Piloten aus. Um ein TA zu berechnen, nutzt ACAS die gleichen Tests wie beim RA (siehe unten), lediglich die Vorwarnzeiten werden um 10 s bis 15 s auf 25 s bis 50 s erhöht. Dem Piloten wird eine akustische Warnung gegeben („Traffic Traffic“) und das betreffende Flugzeug wird auf dem Traffic Display markiert, eine Ausweichenanweisung wird nicht gegeben. Der Pilot soll mit dem TA nur auf das mögliche Entstehen eines RA hingewiesen werden. Nur bei eindeutiger Sicht auf das bedrohende Flugzeug darf er schon jetzt ausweichen [ACAS-Bulletin6].

Traffic Advisories werden nicht koordiniert und auch nicht an Bodenstationen übermittelt, d. h. sie werden nicht über Mode S ausgesendet und können somit nicht direkt ins ACAS-Monitoring durch mithörende Bodenstationen einbezogen werden. Traffic Advisories können lediglich indirekt durch Simulationen auf Flugzeugspuren berechnet werden. Erste Versuche [Pietsch09] anhand beobachteter Bahnen und durch Nachbildung der ACAS-Algorithmen die Traffic-Advisories zu berechnen, wurden bereits durchgeführt, aber für diesen Zweck als noch nicht genügend brauchbar befunden.

Kollisionsdrohung erkennen – RA-Berechnen

ACAS berechnet aus Schrägentfernung r und Relativgeschwindigkeit \dot{r} die Zeit bis zur vorhergesagten Kollision τ_h . Wird diese Zeit kleiner als 15 s bis 35 s (je nach Höhe), so ist die erste Bedingung für ein RA, der Range Test, erfüllt.

$$\tau_0 < \tau_h = \frac{r}{\dot{r}} \quad \text{Horizontales } \tau\text{-Kriterium im Range Test, vereinfacht} \quad (1)$$

Analog bestimmt der Altitude Test, ob die Flugzeuge in den nächsten 15 s bis 35 s die gleiche Höhe erreichen werden.

$$\tau_0 < \tau_v = \frac{\Delta h}{\dot{\Delta h}} \quad \text{Vertikales } \tau\text{-Kriterium im Altitude Test, vereinfacht} \quad (2)$$

Schlägt auch der Altitude Test an, wird ein RA berechnet und an den Piloten ausgegeben.

Da ACAS nur die Annäherung eines Flugzeuges feststellt, nicht aber die genaue Peilung messen kann, kann es auch nur vertikale RAs berechnen, d. h. „Sinken“ oder „Steigen“ anweisen. Eine horizontale Ausweichenanweisung wird nicht gegeben. Das bleibt einem noch zu entwickelnden und zu standardisierenden ACAS III vorbehalten.

Koordination – UF16-30-Resolution-Message

Die ACAS-Geräte in den Flugzeugen sind autonom, d. h. ein ACAS-Gerät erkennt für sich einen Konflikt und berechnet ein RA **nur** für das eigene Flugzeug. Bei Begegnungen zwischen ACAS-ausgerüsteten Flugzeugen kann es durchaus dazu kommen, dass beide ACAS-Geräte Ausweichenanweisungen in die gleiche Richtung erteilen. Um das zu verhindern oder zu korrigieren (siehe Beispiel Event 10 in Abschnitt 9.11), werden die Ausweichmanöver koordiniert.

Sobald ein Flugzeug ein RA berechnet und ausgegeben hat, sendet es eine UF16-30-Resolution-Message (3 in Bild 1) an das bedrohende Flugzeug (Gegner)⁹ und weist damit ein RA-Complement (RAC, Komplement zu dem eigenen Ausweichmanöver) zu. Mit dem RA-Complement wird mitgeteilt, was der Gegner zu unterlassen hat, damit er nicht in die gleiche Richtung ausweicht, z. B. „do not pass above“ oder „do not pass below“. Berechnet ein ACAS also ein „Steigen“-RA so teilt es dem Gegner mit einer UF16-30-Resolution-Message mit: „do not pass above“.

Das bedrohende Flugzeug übernimmt das zugewiesene RA-Complement, speichert es zusammen mit dem Absender und bezieht es in die Auswahl des eigenen RA ein. Über weitere UF16-30 kann ein RAC später widerrufen werden, wenn sich die Situation entspannt hat oder ein anderes RA gewählt wurde. Das RAC dient lediglich der Koordination der ACAS-Geräte, es wird dem Piloten nicht angezeigt.

Die UF16-30-Resolution-Message ist die einzige Koordinationsnachricht, die sowohl Absender als auch Empfänger einer ACAS-Kommunikation enthält. Mit der UF16-30 kann das in dieser Arbeit entwickelte ACAS-Monitoring feststellen, welche Flugzeuge **gemeinsam** an einem ACAS-Eingriff beteiligt sind. Die UF16-30 enthält aber keine direkte Information über das im eigenen Cockpit angezeigte RA.

Quittierung der Koordination – DF16-30-Coordination-Reply

Hat ein Flugzeug ein RAC zugewiesen bekommen, so muss dies durch eine DF16-30-Coordination-Reply-Antwort (4 in Bild 1) quittiert werden, wobei das zugewiesene RAC wiederholt wird. Daneben enthält die DF16-30 den Absender und das eigene RA (rot in Bild 1). ACAS-Monitoring kann somit aus der DF16-30 das angezeigte RA eines Flugzeuges erfahren – aber nur dann, wenn das gegnerische Flugzeug auch ein RA anzeigt und mit einer UF16-30-Resolution-Message zum Melden des RA auffordert.

Meldung an Bodenstationen – UF16-31-RA-Broadcast

Der UF16-31-RA-Broadcast (5 in Bild 1) dient nicht der Koordination, sondern wurde speziell für mithörende Bodenstationen definiert. Vom Beginn bis zur Beendigung eines RA wird ca. alle 8 s ein UF16-31-RA-Broadcast gesendet. Dieser enthält das im Cockpit angezeigte RA und das vom Gegner zugewiesene RAC. Aus der UF16-31 geht nur die Identität des Absenders hervor, der gerade eine RA anzeigt und den UF16-31-RA-Broadcast sendet, der Gegner in der Begegnung wird nicht genannt.

Der UF16-31-RA-Broadcast ist die einzige Nachricht, die bei jedem Ansprechen von ACAS gesendet wird. Dies ist unabhängig davon, ob das andere Flugzeug mit ACAS ausgerüstet ist oder nicht. Aber auch wenn beide Flugzeuge mit ACAS ausgerüstet sind und nur ein ACAS anspricht, ist die UF16-31 die einzige Möglichkeit, das RA mitzuhören. Deshalb muss das ACAS-Monitoring die UF16-31 auswerten. Diese Aufgabe ist nicht trivial, weil die UF16-31 nicht die Mode S-Adresse des Absenders nennt, sondern den Squawk¹⁰, die Kennung des Flugzeuges im Mode A/C-Radar. Die Lösung dieses Problems wird in Abschnitt 6.2.2 erläutert.

⁹ Um die Erklärungen in dieser Arbeit verständlich zu halten, wird „das bedrohende Flugzeug“, vor dem ACAS mit einem Resolution Advisory warnt, als „Gegner“ bezeichnet.

¹⁰ Der Squawk ist die 12-Bit-Flugzeugkennung aus dem bisherigen Mode A/C-Sekundärradar, die von der Flugsicherung dynamisch nach Bedarf (meist per Sprechfunk) einem Flugzeug zugewiesen wird.

2.2 ACAS-Monitoring-Idee

Aus Abschnitt 2.1 folgt nun: Um den Ablauf von ACAS-Eingriffen detailliert zu dokumentieren, müssen die obigen ACAS-Kommunikationen mitgehört werden.

- Die DF11-Squitter liefern die Mode S-Adressen von Flugzeugen im Empfangsbereich des Systems.
- Die DF00-Air-Air-Surveillance-Antworten liefern ein Höhenprofil mit mehr als sekundlichen Updates.
- Durch UF16-30-Resolution-Message wird die Identität des anderen beteiligten Flugzeuges bekannt, sofern es auch mit ACAS ausgerüstet ist.
- Aus DF16-30-Coordination-Reply geht das RA des Absenders der DF16-30 hervor.
- Aus UF16-31-RA-Broadcast kann das RA gelesen werden, auch wenn kein DF16-30-Coordination-Reply von diesem Flugzeug empfangen wurde.

Damit kann durch Mithören der Kommunikation mit Bodenstationen die vertikale Situation vor dem ACAS-Eingriff, das RA und die vertikale Situation nach dem ACAS-Eingriff rekonstruiert werden. Da ACAS nur vertikale Ausweichmanöver anweist, ist das Höhenprofil unerlässlicher Teil der Beschreibung eines ACAS-Eingriffs. Eine zusätzliche Beschreibung der horizontalen Verkehrssituation vervollständigt das Bild. Horizontale Bahndaten können dazu aus Radardaten oder aus sog. ADS-B-Positionsmeldungen (DF17) gewonnen werden. Wie ADS-B-Daten in den Prozess einfließen können, wird in Kapitel 7 erläutert.

Da nun feststeht, welche Nachrichten mitgehört und analysiert werden müssen, ist nun zu klären, wie sie mitgehört werden können. Alle ACAS-Kommunikationen werden über das Mode S-Protokoll abgewickelt. Darum soll in den folgenden Abschnitten das Mode S-Protokoll analysiert werden.

2.3 Mode S-Übertragungsprotokoll

ACAS Systeme messen und kommunizieren mittels spezieller Mode S-Nachrichten auf den Frequenzen des Sekundärradars¹¹ (SSR) 1030 MHz und 1090 MHz. ACAS benutzt den Mode S-Antwortstransponder und gliedert sich mit seinen Aussendungen in das Mode S-Sekundärradarsystem ein. Mode S ist **nicht** dafür konzipiert worden, von einer mithörenden Station analysiert zu werden. Angesichts der starken Übertragungsstörungen, insbesondere im 1090 MHz-Kanal, ist es eine Herausforderung, festzustellen, ob eine mitgehornte Mode S-Nachricht so empfangen wurde, wie diese vom Flugzeug gesendet wurde. Die dazu nötigen Verfahren werden in dieser Arbeit entwickelt und umgesetzt.

¹¹ Ein Sekundärradar (Secondary Surveillance Radar, SSR) ist ein Radar, das mit kooperativen Zielen arbeitet. Im Gegensatz zum Primärradar, bei dem am Ziel nur eine (passive) Reflexion stattfindet, sendet das Sekundärradar ein Datensignal als „Abfrage“, das aktiv vom Transponder im Ziel mit einer „Antwort“ erwidert wird. Wie beim Primärradar wird aus der Laufzeit des Signals und der Richtung der rotierenden Antenne die Position des Ziels ermittelt. In dem Datensignal können des Weiteren beliebige zusätzliche Informationen (z. B. Identität und Höhe des Ziels) übermittelt werden.

2.3.1 Beteiligte Geräte – ACAS, Mode S-Transponder, Mode S-Radar

Mode S ist ein Kommunikationsprotokoll für Sekundärradare. Das Mode S-Protokoll ist als Master-Slave-Protokoll zwischen Radaranlagen und Flugzeugen entwickelt worden. Mode S-Radars senden auf 1030 MHz (Uplink) sog. adressierte Abfragen an Flugzeuge, die Flugzeuge antworten auf 1090 MHz (Downlink). Dazu haben sie einen sog. Mode S-Transponder an Bord. Das „S“ in Mode S steht für selektives Abfragen: Jeder Mode S-Transponder besitzt eine weltweit eindeutige 24-Bit-Adresse, er antwortet nur auf Abfragen an seine Adresse.

ACAS ist ein weiteres Gerät an Bord eines Flugzeuges. Es setzt einen Mode S-Transponder voraus und arbeitet eng mit ihm zusammen. Zur Beobachtung der Umgebung und zur Koordination sendet ACAS wie ein Radar selbst Abfragen an andere Flugzeuge auf 1030 MHz (Uplink) aus und empfängt auf 1090 MHz (Downlink) Antworten. Diese Arbeitsweise von ACAS könnte man als vereinfachtes Mode S-Radar bezeichnen. Im Gegensatz zum Mode S-Radar verwendet ACAS keine rotierende, scharf gebündelte Radar-Antenne, sondern lediglich ein Vier-Antennen-Array.

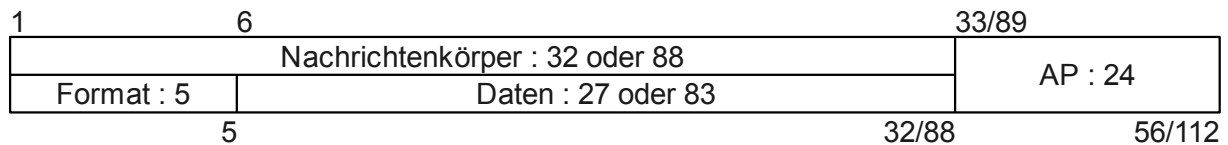
Zum Beispiel sendet ACAS eine UF00-Air-Air-Surveillance-Abfrage auf 1030 MHz (Uplink) adressiert an ein Flugzeug in der Umgebung aus. Der Mode S-Transponder des Abgefragten beantwortet die Abfrage mit einer DF00-Air-Air-Surveillance-Antwort auf 1090 MHz (Downlink). ACAS erhält aus der Antwort selbst die Druckhöhe des Abgefragten, aus der Laufzeit des Abfrage-Antwort-Paares die Entfernung.

2.3.2 Aufbau von Mode S-Nachrichten

Eine Mode S-Abfrage bzw. -Antwort besteht aus der Präambel und der eigentlichen Mode S-Nachricht. Die Präambel als festes Muster dient der Entdeckung des Mode S-Telegramms und der Kompatibilität zum bisherigen Sekundärradarsystem Mode A/C. Die Mode S-Nachrichten in Abfragen (1030 MHz Uplink) werden mit differenzieller Phasenumtastung (DPSK [Proakis]), die Nachrichten in den Antworten (1090 MHz Downlink) werden mit Pulse Position Modulation (PPM [Proakis]) moduliert.

Die demodulierten Mode S-Nachrichten sind im Uplink- und Downlink grundsätzlich gleich aufgebaut. Es gibt kurze und lange Nachrichten, welche 56 Bit bzw. 112 Bit lang sind. Bild 2 zeigt den Aufbau einer Mode S-Nachricht und benutzt dabei die gleiche Notation wie [ICAO Annex10v4]. Oben bzw. unten angestellte Zahlen in Bild 2 bezeichnen Bit-Stellen, im Mode S-Standard laufen Indexe ab „1“. Nach dem Feldnamen wird jeweils die Länge des Feldes in Bit angegeben.

Die ersten 5 Bit einer Nachricht legen das Format (den Typ) einer Nachricht fest. Die folgenden 27 Bit, bzw. 83 Bit bei langen Nachrichten, enthalten die eigentlichen Nutzdaten. Die letzten 24 Bit enthalten das AP-Feld (Address/Parity, bei einigen Formaten als PI-Feld bezeichnet, siehe unten): eine Prüfsumme kombiniert mit einer Adresse. Diese Kombination ist für Netzwerkprotokolle ungewöhnlich und wird gesondert in Abschnitt 2.4 erklärt. Zur Abgrenzung werden die ersten 32 Bit (88 Bit bei langen Nachrichten), der Teil vor dem AP-Feld, auch als Nachrichtenkörper bezeichnet.



*Bild 2: Struktur von Mode S-Abfragen und -Antworten,
Angefügte Zahlen kennzeichnen Länge der Datenfelder bzw. Bitstellen*

Aktuell (im Jahre 2009) sind acht Uplink-Formate und elf Downlink-Formate definiert. Abschnitt 2.5 gibt dazu einen Überblick.

2.4 Ausschluss von Übertragungsfehlern – CRC-Prüfsumme im AP-Feld

ACAS und andere Systeme wie ADS-B senden auf den 1030 MHz- und 1090 MHz-Kanälen, ohne Nachrichtenüberlagerung zu verhindern oder überhaupt zu erkennen. Senden verschiedene Flugzeuge gleichzeitig, überlagern sich die Aussendungen und werden als gestörte Bitfolgen empfangen. Für das Funktionieren von Mode S und für das Mithören ist also eine Möglichkeit zum Ausschluss von Übertragungsfehlern essentiell – dazu besitzt jede Mode S-Nachricht eine 24-Bit-CRC¹²-Prüfsumme. Vor der Verarbeitung einer jeden Nachricht muss deren Prüfsumme getestet werden, um gestörte Bitfolgen auszusondern.

Alle Mode S-Nachrichten enthalten an deren Ende das 24 Bit lange AP-Feld. Das AP-Feld hat eine Doppelfunktion, es enthält gleichzeitig die 24-Bit-Adresse der Nachricht und auch die 24-Bit-Prüfsumme der Nachricht, beide sind mit XOR verknüpft. Diese Verbindung von Adresse und Prüfsumme in einem Datenfeld ist ungewöhnlich für Netzwerkprotokolle und führt zu einigen Herausforderungen bei der Verarbeitung von Mode S-Nachrichten, worauf im Laufe dieser Arbeit immer wieder eingegangen werden muss. In diesem Abschnitt wird die Funktion des AP-Feldes vollständig beschrieben, jedoch wird die Generierung des CRC-Codes nur soweit erklärt, wie zum Verständnis der Arbeit nötig ist. Alle Details zur CRC-Codierung im AP-Feld finden sich in [ICAO Annex10v4].

Senden – AP-Feld und CRC generieren

Bild 3 stellt die CRC-Codierung des AP-Feldes beim Senden einer Nachricht dar. Dabei berechnet der Sender die CRC-Prüfsumme des Nachrichtenkörpers (Nachricht vor AP-Feld). Diese Prüfsumme wird mit dem zu sendenden Inhalt des AP-Feldes verknüpft (per XOR) und die Nachricht mit CRC-codiertem AP-Feld gesendet. Die meisten Formate enthalten im AP-Feld die Mode S-Adresse, für diese ist das CRC-codierte AP-Feld demnach die Adresse XOR der CRC-Prüfsumme.

¹² CRC: engl. cyclic redundancy check, dt. zyklische Redundanzprüfung ist eine Klasse von Verfahren zur Bestimmung eines Prüfwertes für Daten, um Fehler bei der Übertragung erkennen zu können.

Nachricht:

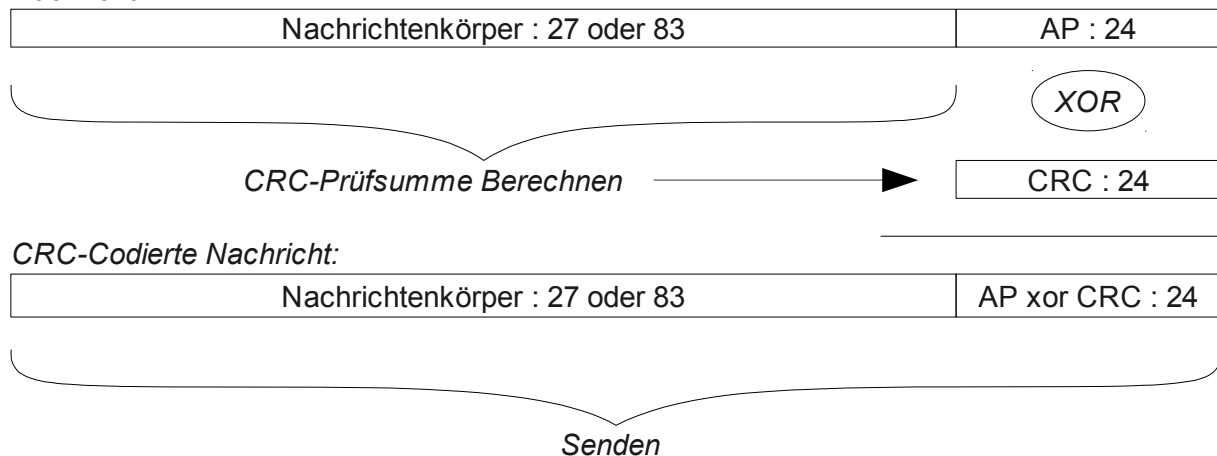


Bild 3: CRC-Codierung beim Senden von Mode S-Nachrichten

Empfang – AP Feld decodieren

Bild 4 zeigt die Decodierung des AP-Feldes einer Nachricht beim Empfang. Der Empfänger berechnet die Prüfsumme des empfangenen Nachrichtenkörpers und verknüpft diese mit dem empfangenen CRC-codierten AP-Feld (auch XOR¹³). Ist die Nachricht störungsfrei empfangen worden, ist dieser Rest das ursprüngliche AP-Feld vor dem Senden. Wurde die Nachricht gestört, enthält dieser Rest eine gestörte Bitfolge. Der Empfänger nimmt eine Nachricht als korrekt an, wenn dieser Rest (der Inhalt des AP-Feldes ohne die Prüfsumme) die eigene Mode S-Adresse, die Mode S-Adresse eines Kommunikationspartners oder eine feste Bitfolge mit besonderer Bedeutung ist.

Für jede empfangene Nachricht muss das AP-Feld decodiert werden. Zur Vereinfachung bezeichnet daher der Ausdruck „Inhalt des AP-Feldes“ im Folgenden immer den decodierten Inhalt des AP-Feldes.

Empfangene Nachricht:

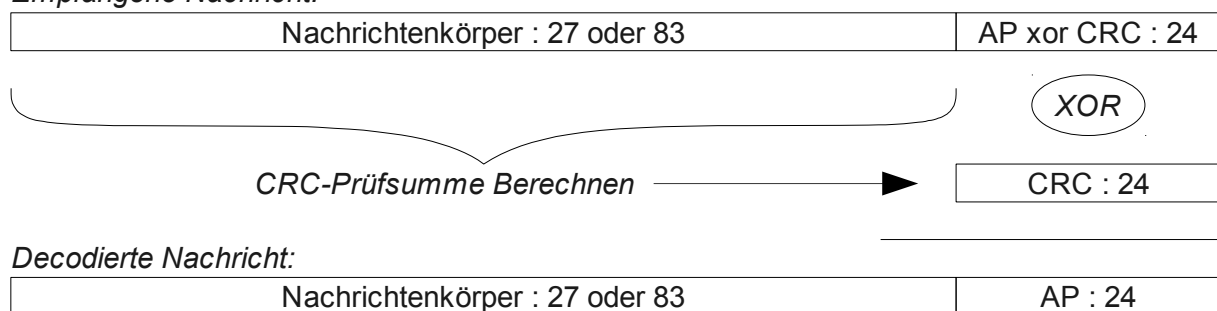


Bild 4: CRC-Decodierung beim Empfang von Mode S-Nachrichten

Inhalt des AP-Feldes

In Uplink-Nachrichten enthält das AP-Feld die Adresse des Empfängers der Nachricht. Ein Mode S-Transponder nimmt eine Nachricht als „an ihn gerichtet“ an, wenn seine eigene Mode S-Adresse nach der CRC-Decodierung im AP-Feld steht.

¹³ XOR mit gleichem Argument ist selbstinvers, d. h. $(a \text{ XOR } b) \text{ XOR } b = a$

In Downlink-Nachrichten enthält das AP-Feld die Adresse des Absenders der Nachricht. Ein Mode S-Radar oder ein ACAS nimmt eine Mode S-Downlink-Nachricht dann an, wenn das AP-Feld eine Adresse enthält, an die es gerade eine Abfrage gesendet hat.

An alle Flugzeuge gerichtete Uplink-Nachrichten enthalten im AP-Feld die feste Bitfolge FFFFFFFF_{hex}¹⁴. FFFFFFFF_{hex} wird von [ICAO Annex10v4] je nach Format als „All-Call-Adresse“ oder „Broadcast-Adresse“ bezeichnet.

Downlink-Nachrichten in den Formaten DF11, DF17 und DF18 (sog. Squitter) enthalten im AP-Feld 000000_{hex}¹⁵, weil diese ohne zusätzliches Wissen decodiert werden sollen. Das AP-Feld der Formate DF11, DF17 und DF18 wird im [ICAO Annex10v4] PI-Feld genannt. Um im Folgenden die gleichen Bitstellen nicht in einem Satz einmal als AP- und einmal als PI-Feld bezeichnen zu müssen, wird in dieser Arbeit die Bezeichnung AP-Feld für die letzten 24 Bit einer Nachricht verwendet, wenn eine Aussage auf beide Bezeichnungen des Feldes zutrifft.

2.4.1 Prüfung des AP-Feldes durch einen passiven Mode S-Beobachter

Ein passiver Mode S-Beobachter, wie der ACAS-Monitor, empfängt von Flugzeugen und Radaren ausgesendete Nachrichten, sendet aber selbst keine Abfragen aus. Auch hier müssen die beobachteten Mode S-Nachrichten auf Übertragungsfehler überprüft werden. Nachrichtenformate, bei denen das AP-Feld die bekannte Summe 000000_{hex} oder FFFFFFFF_{hex} enthält, können direkt geprüft werden. Komplizierter ist es bei Nachrichten, die im AP-Feld eine Flugzeugadresse enthalten. Beim Empfangen einer solchen Nachricht stellt sich die Frage, ist das AP-Feld die Adresse eines wirklichen Flugzeuges im Luftraum oder ist das AP-Feld durch eine Störung entstanden.

Die Lösung bilden die bereits erwähnten DF11-Squitter. Diese werden von Flugzeugen regelmäßig ca. jede Sekunde ausgesendet. Sie enthalten 000000_{hex} im AP-Feld und können somit ohne Vorwissen geprüft werden. Im Nachrichtenkörper enthalten sie die Absenderadresse.

Der in dieser Arbeit entwickelte ACAS-Monitor erfährt durch Mithören der DF11-Squitter (und DF17, DF16-30, die ähnlich aufgebaut sind, aber nicht von jedem Mode S-Transponder gesendet werden) die Mode S-Adressen aller gerade im Empfangsbereich befindlichen Flugzeuge – diese werden in einer Liste abgelegt. Gegen diese Liste müssen die AP-Felder der anderen eingehenden Nachrichten geprüft werden. Die Liste sollte alle Flugzeuge enthalten, die sich gerade im Empfangsbereich befinden, damit keine von diesen Flugzeugen störungsfrei empfangene Mode S-Nachricht verworfen wird. Zum anderen sollte die Liste durch regelmäßige Bereinigung aber auch klein gehalten werden, damit gestörte Nachrichten kein gültiges AP-Feld (gültige Adresse) vortäuschen können und gestörte Nachrichten als „korrekt übertragen“ angenommen werden.

¹⁴ FFFFFFFF_{hex} ist die in dieser Arbeit verwendete Schreibweise für hexadezimale Zahlen.

¹⁵ Bei DF11-Antworten auf Radar-Abfragen, wird in den letzten 7 Bit des AP-Feldes noch die Identität des Radars übermittelt.

2.5 Aufbau der Mode S-Formate im Überblick

In den vorausgehenden Abschnitten wurde der Aufbau der Mode S-Nachrichten dargestellt. Dieser Abschnitt (2.5) und die Abschnitte 2.6, 2.7 und 2.8 schaffen einen Überblick über den Aufbau jedes einzelnen Mode S-Formates. Damit wird gezeigt, wie die ACAS-Kommunikationen aufgebaut sind und welche weiteren Formate von dem zu entwickelnden ACAS-Monitoring ausgewertet werden müssen.

Zur Zeit (2009) sind die acht Uplink Formate UF00, UF04, UF05, UF11, UF16, UF20, UF21 und UF24 standardisiert, Bild 5 auf Seite 15 zeigt diese und ihren schematischen Aufbau. Die Downlinkformate DF00, DF04, DF05, DF11, DF16 bis DF21 und DF24 sind ebenfalls standardisiert und werden in Bild 6 dargestellt.

Format Kurze Uplink-Nachricht – 56 Bit

UF00	00000	:3	RL:1	:4	AQ:1	:18	AP:24
UF04	00100	PC:3	RR:5	DI:3	SD:16	AP:24	
UF05	00101	PC:3	RR:5	DI:3	SD:16	AP:24	
UF11	01011	PR:4	IC:4	CL:3	:16	AP:24	

Format Lange Uplink-Nachricht – 112 Bit

UF16	10000	:3	RL:1	:4	AQ:1	:18	MU:56	AP:24
UF20	10100	PC:3	RR:5	DI:3	SD:16	MA:56	AP:24	
UF21	10101	PC:3	RR:5	DI:3	SD:16	MA:56	AP:24	
UF24	11	RC:2	NC:4	MC:80	AP:24			

Bild 5: Mode S-Uplink-Formate

Format Kurze Downlink-Nachricht – 56 Bit

DF00	00000	VS:1	:2	SL:3	:2	RI:4	:2	AC:13	AP:24
DF04	00100	FS:3	DR:5	UM:6	AC:13	AP:24			
DF05	00101	FS:3	DR:5	UM:6	ID:13	AP:24			
DF11	01011	CA:3	AA:24	PI:24					

Format Lange Downlink-Nachricht – 112 Bit

DF16	10000	VS:1	:2	SL:3	:2	RI:4	:2	AC:13	MV:56	AP:24
DF17	10001	CA:3	AA:24	ME:56	PI:24					
DF18	10010	CF:3	AA:24	ME:56	PI:24					
DF19	10011	AF:3	:104							
DF20	00100	FS:3	DR:5	UM:6	AC:13	MB:56	AP:24			
DF21	00101	FS:3	DR:5	UM:6	ID:13	MB:56	AP:24			
DF24	11	:1	KE:1	ND:4	MD:80	AP:24				

Bild 6: Mode S-Downlink-Formate

2.6 Mode S-Formate des Mode S-Radars

Die Bodenstationen des in dieser Arbeit entwickelten ACAS-Monitors empfangen zwangsläufig auch Mode S-Radarkommunikation. Auch wenn die Mode S-Radar-Kommunikationen nicht direkt ACAS-relevant sind, lassen sich zusätzliche Informationen für das ACAS-Monitoring gewinnen.

Das Mode S-Radar benutzt die Formate UF04/DF04, UF05/DF05, UF11/DF11, UF20/DF20, UF21/DF21 und UF24/DF24 (Bild 5 und Bild 6).

Das Mode S-Radar akquiriert in den Empfangsbereich einfliegende Flugzeuge mit UF11-All-Call-Abfragen an alle Flugzeuge im Radarstrahl. Flugzeuge die noch nicht von diesem Radar abgefragt wurden, antworten auf den All-Call mit einer DF11-All-Call-Antwort, die (fast) identisch zum DF11-Squitter ist und vom ACAS-Monitor genauso verarbeitet werden kann.

Das Radar fragt mit einer UF04 die Druckhöhe des Flugzeuges ab, im AP-Feld der UF04 steht dabei die Adresse des abgefragten Flugzeuges. Als Antwort sendet das Flugzeug eine DF04, in der das AC-Feld die Druckhöhe enthält (Auflösung 25 ft, bei älteren Druckhöhenmessern 100 ft) und das AP-Feld die Adresse des Flugzeuges beinhaltet. Mitgehörte DF04 können dem ACAS-Monitor Höheninformationen über Flugzeuge liefern. Analog wird mit UF05 der Squawk des Flugzeuges abgefragt. Die UF05 wird mit einer DF05 beantwortet, welche dann den Squawk im Nachrichtenkörper und die Mode S-Adresse im AP-Feld enthält. Aus den DF05 kann der ACAS-Monitor den aktuell einer Flugzeug-Mode S-Adresse zugeordneten Squawk ermitteln.

Zusätzlich zu Druckhöhe und Squawk kann das Mode S-Radar noch eine Reihe weiterer Daten an Flugzeuge senden oder abfragen. Dazu dienen die langen Abfragen UF20, UF21 bzw. die Antworten DF20, DF21 darauf. Die UF20 ist einfach eine um 56 Bit verlängerte Version der UF04, analog verhalten sich UF21 zu UF05, DF20 zu DF04 und DF21 zu DF05. Sollen zusätzliche Informationen (sog. COMM-A oder COMM-B) übertragen werden, werden anstelle der kurzen Nachrichten die entsprechenden langen Versionen übertragen. Die MA bzw. MB Felder enthalten dann die zusätzlichen Daten. Diese Zusatzinformationen sind z. B. Flugplannummern oder auch ACAS-RA-Reports [ICAO Annex10v4].

UF24/DF24 sind für sogenannte „Extended Length Communication (ELM)“ [ICAO Annex10v4] vorgesehen, sie sollen als Netzwerk-Transportschicht für höhere Protokolle dienen. Sie werden aber bisher kaum benutzt und spielen in dieser Arbeit daher keine Rolle.

2.7 Codierung der ACAS-Informationen im Mode S-Protokoll

ACAS benutzt die Mode S-Formate UF00/DF00 und UF16/DF16. Zu Beginn dieses Kapitels wurde die Funktion der ACAS-Formate beschrieben – hier wird nun noch deren Aufbau beschrieben, da die ACAS-Formate für diese Arbeit von zentraler Bedeutung sind. Es wird gezeigt, wie die ACAS-Kommunikationen in Mode S-Nachrichten codiert werden und welche Felder vom ACAS-Monitor interpretiert werden müssen.

Der UF00-Air-Air-Altitude-Request (Bild 5) enthält die Adresse des Abgefragten im AP-Feld. Die UF00-Abfrage wird mit einer DF00-Air-Air-Altitude-Reply (Bild 6), die die Druckhöhe im Nachrichtenkörper und die Adresse des Abgefragten im AP-Feld enthält, beantwortet.

Komplexer ist der Aufbau der UF16/DF16-Formate (Bild 7). Sie sind lange Versionen der UF00/DF00, wobei die ACAS-Informationen in den angehängten MU bzw. MV-Feldern codiert werden.

UF16-30	Format	UF00	MU:56										AP
	Felder	UDS	-	MTB	CVC	VRC	CHC	HRC	-	HSB	VSB	MID	
	10000 :5	30 :29	:2	:1	:1	:2	:2	:3	:3	:3	:5	:4	:24

UF16-31	Format	UF00	MU:56									AP
	Felder	UDS	ARA		RAC	RAT	MTE	-	AID	CAC		
	10000 :5	31 :29	:2	:18		:3	:1	:1	:2	:13	:13	FFFFFF :24

UF16-32	Format	UF00	MU:56										AP
	Felder	UDS	-----									MID	
	10000 :5	32 :29	:2	:24									FFFFFF :24

DF16-30	Format	DF00	MV:56								AP
	Felder	VDS	ARA		RAC	RAT	MTE	-----			
	10000 :5	30 :29	:2	:18		:3	:1	:1	:28		

Bild 7: ACAS-Nachrichten UF16 und DF16

2.7.1 UF16-30-Resolution-Message

Die UF16-30-Resolution-Message ist Teil der Koordination zwischen ACAS-Flugzeugen. Eine Koordination und die damit verbundenen Nachrichten werden nur gesendet, wenn beide Flugzeuge mit ACAS ausgerüstet sind.

Mit einer UF16-30-Resolution-Message weist das ACAS-Gerät, das einen Konflikt erkannt hat, seiner Bedrohung ein RA-Complement zu. Das RA-Complement, das im VRC-Feld der UF16-30 übertragen wird, übermittelt kein RA, sondern führt nur die Koordination aus. Berechnet ein ACAS also ein „Steigen“-RA, so teilt es der Bedrohung mit einer UF16-30-Resolution-Message mit: „do not pass above“. Damit wird eine Forderung der ACAS-Standardisierung implementiert: Das ACAS-Gerät an Bord eines Flugzeuges ist autonom, d. h. es erteilt anderen Flugzeugen keine Anweisungen und nimmt von anderen Flugzeugen keine Anweisung entgegen. Die autonomen ACAS-Geräte koordinieren lediglich mittels RA-Complement in der UF16-30-Resolution-Message die eigenen RAs untereinander.

Über das CVC-Feld in der UF16-30 kann ein zuvor in einer anderen UF16-30 ausgesendetes RA-Complement später widerrufen werden. Das wird zum Beispiel nötig, wenn sich die Situation entspannt hat oder ein anderes RA-Complement gewählt wurde. Um Übertragungsfehler bei dieser sicherheitsrelevanten Kommunikation auszuschließen, sind VRC und CVC durch eine 4-Bit-Prüf-

summe im VSB-Feld noch einmal zusätzlich gesichert. Das MTB-Feld wird gesetzt, wenn mehr als ein bedrohendes Flugzeug bearbeitet wird. Die Felder CHC, HRC und HSB sind für horizontale Ausweichmanöver eines zukünftigen ACAS III vorbehalten und spielen hier keine Rolle.

Wie bereits erwähnt, ist die UF16-30 die einzige Nachricht, die sowohl Absender (MID-Feld) als auch Empfänger (AP-Feld) einer ACAS Kommunikation enthält. Nur aus der UF16-30 kann das ACAS-Monitoring feststellen, welche Flugzeuge **gemeinsam** an einem ACAS-Eingriff beteiligt sind.

2.7.2 DF16-30-Coordination-Reply

Die UF16-30-Resolution-Message wird durch einen DF16-30-Coordination-Reply bestätigt. Die DF16-30 wiederholt das zugewiesene RA-Complement im gleichnamigen RAC-Feld. Weiterhin enthält die Nachricht den Absender (AP-Feld) und das RA (ARA-Feld) des Koordinierten. Ein gesetztes RAT-Bit zeigt das Ende eines RA an. Ein gesetztes MTE-Bit zeigt an, dass mehrere Flugzeuge gleichzeitig als Bedrohung angesehen werden. Aus der DF16-30 geht nur die Identität des Koordinierten und sein RA (falls vorhanden) hervor; der Koordinierende wird nicht genannt.

2.7.3 UF16-31-RA-Broadcast

Der UF16-31-RA-Broadcast wurde speziell für mithörende Bodenstationen definiert. Er enthält:

- das im Cockpit angezeigte RA (ARA-Feld)
- das vom Gegner erhaltene RA-Complement (RAC-Feld)
- das RA Terminated Bit (RAT-Feld), zeigt das Ende eines RA an
- das Multiple Threat Encounter Bit (MTE-Feld)
- den Squawk des Absenders (AID-Feld)
- die Druckhöhe des Absenders (CAC-Feld)
- das AP-Feld (FFFFFF_{hex})

Aus der UF16-31 geht nur die Identität des Absenders hervor, der gerade eine RA anzeigt, der Gegner wird nicht genannt. Die UF16-31 enthält nicht die Mode S-Adresse des Absenders, sondern den Squawk, die Kennung des Flugzeuges im Mode A/C-Radar.

Die UF16-31 wurde in den 1990er Jahren für eine Flugsicherung standardisiert, die nur über Mode A/C-Radar verfügte und Flugzeuge anhand des Squawks identifizierte. Der in dieser Arbeit entwickelte Ansatz eines umfassenden ACAS-Monitorings, das Koordination und RA-Broadcast gemeinsam betrachtet, war damals nicht angedacht. Sollen aber Daten aus diversen Mode S-Übertragungen zusammengeführt werden, muss jeder UF16-31 eine Mode S-Absenderadresse zugeordnet werden – der ACAS-Monitor muss wissen, welcher Squawk welcher Mode S-Adresse gerade zugeordnet ist.

Der Squawk wird von der Flugsicherung dynamisch nach Bedarf vergeben. Eine Flugsicherung die mit Mode S arbeitet hat also grundsätzlich Informationen zum Zusammenhang zwischen Squawk und Mode S-Adresse. Diese Informationen müssen dem ACAS-Monitoring zugänglich gemacht werden. Dies wurde in dieser Arbeit dadurch gelöst, dass empfangene DF05-Antworten

auf Mode S-Radarabfragen mitgehört werden und damit Squawk und Mode S-Adressen einander zugeordnet werden können. Damit ist das hier vorgestellte ACAS-Monitoring auf die Existenz eines Mode S-Radars im Empfangsbereich angewiesen. Sollte ACAS-Monitoring in Gebieten ohne Mode S-Radar eingesetzt werden, so müsste die Flugsicherung Informationen zu den zugewiesenen Squawks bereitstellen.

Squawk in UF16-31 nicht in Mode A-Reihenfolge

Der RA-Broadcast enthält den Squawk des Absenders im AID-Feld. Laut [ICAO Annex10v4] müssen die Bits des Squawk im UF16-31-RA-Broadcast wie in Mode A SSR-Antworten angeordnet werden. Stattdessen ordneten **alle** beobachteten Flugzeuge die Identity-Bits nach normaler binärer Gewichtung. Auch [RTCA DO185A] gewichtet die Bits binär, ebenso behält der Entwurf des Nachfolgestandards [RTCA DO185B] diesen Fehler bei. Hier herrscht offensichtlich eine Diskrepanz zwischen den ICAO und RTCA Standards.

Beispiel:

Am 14. Dezember 2005 wurde der folgende RA-Broadcast empfangen (Bit 63 bis 75 unterstrichen):

8080000031C00001D7E9007FE5C3_{hex}

Bits 63 bis 75 enthalten den Squawk. Der nach [ICAO Annex10v4] codierte Squawk ist 7727. Kein Flugzeug hat zu dieser Zeit diesen Squawk gemeldet. Werden die Bits des empfangenen Squawks jedoch nach binärer Gewichtung decodiert, so ergibt sich ein Squawk von 3577, der tatsächlich zu einem beobachteten Flugzeug gehörte.

Bits die im RA-Broadcast empfangen wurden:

63						
...	011	101	0	111	111	...
75						

Anordnung nach [ICAO Annex10v4] (wie Mode A):

63						
...	C ₁ A ₁ C ₂	A ₂ C ₄ A ₄	0	B ₁ D ₁ B ₂	D ₂ B ₄ D ₄	...
75						

sortiert nach [ICAO Annex10v4] zu:

A ₄	A ₂	A ₁	B ₄	B ₂	B ₁	C ₄	C ₂	C ₁	D ₄	D ₂	D ₁	
1	1	1	1	1	1	0	1	0	1	1	1	= <u>7727</u>

Binär gewichtete Bits (nach RTCA-DO185A):

0	1	1	1	0	1	1	1	1	1	1	1	= <u>3577</u>
---	---	---	---	---	---	---	---	---	---	---	---	---------------

Der nicht [ICAO Annex10v4]-konform codierte Squawk 3577 konnte einem Flugzeug zugeordnet werden, das zu dieser Zeit weitere RA-Aktivitäten zeigte.

Das hier entwickelte ACAS-Monitoring nimmt beide Squawks als die Absender der Nachricht und sucht mit Ihnen nach passenden Mode S-Adressen. Dadurch wird der ohnehin nicht immer eindeutige Squawk noch häufiger mehrdeutig. Für den ACAS-Monitor werden im Rahmen dieser Arbeit spezielle Routinen entwickelt, die damit umgehen (siehe Abschnitt 6.2.2).

Diese Diskrepanz der Standards wurde einem Mitglied des zuständigen ICAO Aeronautical Surveillance Panel (ASP) mitgeteilt.

2.7.4 UF16-32-ACAS-Broadcast

Da hier alle UF16/DF16 Formate aufgezählt wurden, sei auch der UF16-32-ACAS-Broadcast erwähnt. Dieser wird von ACAS-Flugzeugen alle 8 s bis 10 s ausgesendet, **unabhängig** davon, ob ein RA vorliegt. Er enthält die eigene Mode S-Adresse im Nachrichtenkörper und im AP-Feld FFFFFFFF_{hex}. Damit ähnelt der UF16-32 ACAS-Broadcast in Funktion und Aufbau eher einem DF11-Squitter, mit der Besonderheit, dass er im Uplink gesendet wird. Damit kann ACAS-Monitoring den UF16-31 ACAS-Broadcast als weiteren Squitter nutzen, um auch aus dem Uplink Mode S-Adressen zu akquirieren.

2.7.5 Codierung des Resolution Advisories in ACAS-Nachrichten

Das Resolution Advisory (RA) unterscheidet nicht nur zwischen „Steigen“- und „Sinken“, sondern ACAS differenziert feiner, weshalb der ACAS-Monitor auch diese Differenzierung übernehmen muss. Welches RA gerade aktiv ist, teilt ACAS im ARA-Feld des UF16-31-RA-Broadcast und im ARA-Feld der UF16-30-Resolution-Message mit. Der ACAS-Monitor muss dieses Feld auswerten, um Fragen nach:

- Standardkonformität der Aussendungen,
- Angemessenheit der ACAS-Anweisung,
- Korrektheit und Schnelligkeit der Reaktion des Flugzeuges

beantworten zu können.

ACAS unterscheidet RAs nach Richtung, Steigrate, Steigratenänderung, usw. Alle Eigenschaften und die entsprechenden ARA-Bits sind im Folgenden kurz dargestellt.

Multiple Threat Encounters (MTE-Bit)

ACAS unterscheidet zwischen Begegnungen mit einer Bedrohung (Single Threat) und Begegnungen, in denen mehreren Bedrohungen gleichzeitig ausgewichen werden muss (Multiple Threat Encounters, MTE). Multiple Threat Encounters wurden im Laufe der Langzeitbeobachtung (Kapitel 9) nicht beobachtet und werden in dieser Arbeit daher nicht weiter betrachtet. Die entwickelte Software kann aber sehr wohl Multiple Threat Encounters erkennen und dokumentieren.

No RA (ARA-Bit 1¹⁶)

Der einfachste Zustand des ACAS ist, kein RA zu haben, dieser wird mit ARA-Bit 1 = 0 mitgeteilt.

¹⁶ Wie bei allen Feldern im [ICAO Annex10v4] beginnen die Indizes bei 1

Preventive RA / Corrective RA (ARA-Bit 2)

Ein Corrective RA (ARA-Bit 2 = 0) fordert zur Änderung der vertikalen Rate auf, während ein Preventive RA (ARA-Bit 2 = 1) lediglich vor dem Einnehmen einer bestimmten vertikalen Rate warnt [ICAO Annex10v4]. Die „Steigen“-Anweisung eines ACAS kann also durchaus ein Preventive RA sein, wenn das Flugzeug bereits steigt.

Richtung des RA – Upward Sense / Downward Sense (ARA-Bit 3)

Richtung (ARA-Bit 3) und Stärke (ARA-Bits 7 und 8, siehe unten) von RAs sind in Tabelle 1 systematisiert zusammengefasst. Daneben ist auch die Kodierung der ARA-Bits dargestellt.

Resolution Advisory, RA (Ausweichanweisung)	Bezeichnung der Bits im ARA-Feld	ARA-Bits 3 7 4
Nach oben gerichtetes RA	Upward Sense RA	0
Nicht stärker sinken als -2000 ft/min	Upward, Vertical Speed Limit	0 0 0
Nicht stärker sinken als -1000 ft/min		
Nicht stärker sinken als -500 ft/min		
Nicht sinken 0 ft/min		
Steigen +1500 ft/min	Upward, Positive	0 1 0
Verstärktes Steigen +2500 ft/min	Upward, Positive, Increased Rate	0 1 1
Nach unten gerichtetes RA	Downward Sense RA	1
Nicht stärker steigen als +2000 ft/min	Downward, Vertical Speed Limit	1 0 0
Nicht stärker steigen als +1000 ft/min		
Nicht stärker steigen als +500 ft/min		
Nicht steigen 0 ft/min		
Sinken -1500 ft/min	Downward, Positive	1 1 0
Verstärktes Sinken -2500 ft/min	Downward, Positive, Increased Rate	1 1 1

Tabelle 1: Angewiesene Steig- bzw. Sinkraten bei ACAS RAs

ACAS teilt die RAs mittels ARA-Bit 3 in Upward Sense RAs und Downward Sense RAs ein. Beim **Upward** Sense RA (ARA-Bit 3 = 0) weist ACAS eine Ausweichbewegung nach **oben** an (z. B. „Steigen“) – oder fordert zumindest zum Halten einer Vertikalgeschwindigkeit **oberhalb** eines bestimmten Wertes auf (z. B. „Nicht stärker Sinken als -2000 ft/min“).

Beim **Downward** Sense RA (ARA-Bit 3 = 1) weist die Ausweichbewegung nach **unten** oder die Vertikalgeschwindigkeit muss zumindest **unter** einer Schwelle bleiben.

Stärke des RA – Vertical Speed Limit / Positive (ARA-Bit 7) und Positive Increased Rate (ARA-Bit 4)

ARA-Bit 7 und ARA-Bit 4 beschreiben gemeinsam die Stärke eines RA. Diese Bits sind im Zusammenhang mit dem „Upward Sense / Downward Sense (ARA-Bit 3)“ zu betrachten, da dieses die Richtung vorgibt.

Zum besseren Verständnis wird zunächst das Upward Sense RA betrachtet. ACAS unterscheidet bei Upward Sense RA zwischen sechs Stärken: „Nicht stärker sinken als -2000 ft/min“, „Nicht stärker sinken als -1000 ft/min“, „Nicht stärker sinken als -500 ft/min“, „Nicht sinken“, „Steigen“ und „Verstärktes Steigen“ (vgl. Tabelle 1 obere Hälfte). Die schwächste Anweisung ist „Nicht stärker sinken als -2000 ft/min“, je nach Bedarf wählt ACAS eine stärkere Anweisung. Die Anweisungen „Nicht stärker sinken als ...“ und „Nicht Sinken“ werden als Vertical Speed Limits (VSL) bezeichnet (ARA-Bit 7 = 0). Die Anweisungen „Steigen“ und „Verstärktes Steigen“ werden als positive RA bezeichnet (ARA-Bit 7 = 1). Wobei „Verstärktes Steigen“ als Positive Increased Rate RA (ARA-Bit 7 = 1 und ARA-Bit 4 = 1) bezeichnet wird.

Im Gegensatz zur unterschiedlichen Codierung von „Steigen“ und „Verstärktem Steigen“, werden die VSL-RAs alle gleich codiert. Eine Unterscheidung der VSL-RAs ist in der ACAS-Kommunikation nach dem ACAS-Standard nicht vorgesehen. Das ACAS-Monitoring kann folglich die VSL-RAs auch nicht unterscheiden. Im zukünftigen ACAS II Version 7.1 [RTCA DO185B] werden alle VSL-RAs zu „Nicht Sinken“ bzw. „Nicht Steigen“ vereinfacht zusammengefasst.

Downward Sense RAs (vgl. Tabelle 1 untere Hälfte) werden genauso in VSL-RAs, Positive-RAs und Positive-Increased-Rate-RAs unterschieden.

Entgegen der Darstellung in verschiedenen Literaturquellen ist ein Vertical Speed Limit RA nicht automatisch Preventive. „Vertikal Speed Limit“ bzw. „Positive“ zum einen und „Preventive“ bzw. „Corrective“ zum anderen sind unabhängige Eigenschaften eines RA.

Sense Reversal (ARA-Bit 5)

ACAS kontrolliert sekundlich, ob das angewiesene RA noch der veränderten Situation entspricht oder ein anderes RA angewiesen werden muss. Dies kann eintreten, wenn sich eines der Flugzeuge nicht wie linear vorhergesagt verhält oder das ACAS-Gerät der Bedrohung ein inkompatibles RA ausgewählt hat. In solchen Fällen kann ACAS das RA abschwächen, verstärken oder ganz umkehren. Muss ACAS das RA umkehren, so wird ein solches RA als Sense Reversal bezeichnet und ARA-Bit 5 auf 1 gesetzt. Auf ein Sense Reversal werden auch die Piloten gesondert hingewiesen.

Altitude Crossing (ARA-Bit 6)

ACAS kann ein Ausweichmanöver anweisen, bei dem sich die Höhen beider Flugzeuge kreuzen. Diese RAs zwingen den Piloten zu der Bedrohung hin zu manövrieren [ICAO Annex10v4]. Die ACAS-Logik versucht ohne Altitude Crossing RAs auszukommen, kann sie aber nicht ganz vermeiden. Die Diskussion, ob Altitude Crossing RAs zu unterbinden sind oder nicht, ist nicht Gegenstand dieser Arbeit. Die Ergebnisse der in dieser Arbeit durchgeführten Langzeitbeobachtung von ACAS-Eingriffen (Kapitel 9) können aber sehr wohl benutzt werden, um Aussagen zu Altitude Crossing RAs zu machen.

2.8 Selbsttätige Aussendungen der Flugzeuge – Extended Squitter – ADS-B

Über die Mode S-Kanäle werden neben der ACAS-Überwachung, ACAS-Koordination und der Radar-Flugzeug-Kommunikation auch ADS-B-Daten übertragen. Bei ADS-B (Automatic Dependent Surveillance – Broadcast) senden die Mode S-Transponder der Flugzeuge regelmäßig und

selbstständig (d. h. ohne Anfrage) Zustandsdaten aus. Diese umfassen z. B. aktuelle Position, Flughöhe, Flugrichtung, Geschwindigkeit, Flugzeugtyp und weitere Informationen. Die ADS-B-Daten werden in DF17-Extended-Squitter-Nachrichten übertragen und werden somit beim ACAS-Monitoring mit empfangen.

ADS-B ist in einigen Ländern, jedoch nicht in Deutschland, Pflichtausrüstung. Dennoch sind etwa 60% (Abschnitt 4.3.2) aller Mode S-ausgerüsteten Flugzeuge mit ADS-B ausgerüstet. Besonders interessant sind ADS-B-Positions-Meldungen, da damit die Position von Flugzeugen in das ACAS-Monitoring eingebracht werden kann, ohne dass Radardaten einbezogen werden müssen. Informationen zu ACAS-RAs werden (leider) nicht per ADS-B ausgesendet, entsprechende Anträge wurden während der ADS-B-Standardisierung abgelehnt [Orlando01].

Neben den oben genannten Zustandsdaten werden in jeder DF17 auch die Felder der DF11 übertragen und im PI-Feld der DF17 wird 000000_{hex} gesendet. Damit kann die DF17 wie eine DF11 ohne Vorwissen geprüft werden und kann somit dem ACAS-Monitor zum Akquirieren von Mode S-Adressen dienen. Wie ADS-B in das ACAS-Monitoring eingebunden wird, wird in Kapitel 7 erklärt.

2.9 RA-Abfrage mittels Mode S-Radar – RA-Downlink

Mode S-Radare können mittels sog. RA-Downlink Nachrichten (entweder DF20-30 oder DF21-30) das gerade angezeigte RA eines Flugzeuges abfragen.

Ist in einem Flugzeug gerade ein RA aktiv oder war in den letzten 18 s ein RA aktiv, so zeigt das Flugzeug dies einem Mode S-Radar an. Das Radar kann dann das RA vom Flugzeug abfragen. Je nach Abfrage antwortet das Flugzeug entweder mit einer DF20-30 oder mit einer DF21-30. Die DF20-30 bzw. DF21-30 ist ein um den RA-Downlink verlängerter DF04-Altitude-Reply bzw. DF05-Identify-Reply. Ob der RA-Downlink in einer DF20-30 oder DF21-30 übermittelt wird, hängt nur davon ab, ob das Radar den RA-Downlink zusammen mit der Höhe oder dem Squawk abgefragt hat.

Der RA-Downlink enthält die Mode S-Adresse des Absenders, das RA (ARA-Feld), den RA Terminated Indicator (RAT-Feld) und die Mode S-Adresse des Gegners (bzw. dessen Squawk, Abstand und Richtung, wenn der Gegner nicht mit Mode S-ausgerüstet ist). Der RA-Downlink enthält nicht den genauen Zeitpunkt des RA.

Sowohl Eurocontrol als auch MIT-Lincoln-Lab experimentierten mit dem RA-Downlink ([EurocontrolPASS] bzw. [DrummKuchar07]). Dabei wurden die RA-Downlinks mit einzelnen Mode S-Radare über Zeiträume bis zu mehreren Monaten hinweg abgefragt und ausgewertet. Die Anzahl der dabei täglich beobachteten ACAS-Events lag in ähnlichen Größenordnungen wie bei der Dauerbewertung für diese Arbeit (siehe Abschnitt 9.20).

Die Aufzeichnung und Analyse von RA-Downlinks ist für den Betreiber von Mode S-Radaranlagen naheliegend und durchaus zweckmäßig. Sie kann aber nicht als ACAS-Monitoring im Sinne dieser Arbeit angesehen werden, denn der RA-Downlink ist nur die Abfrage des RAs, nicht die Beobachtung des eigentlichen Vorganges der Messung der Höhen (DF00) und Koordination des RAs (UF16-30, UF16-30, UF16-31). Mit dem RA-Downlink kann somit die Funktion des ACAS nur indirekt beobachtet werden. Außerdem ist der RA-Downlink nicht zeitgenau, sondern das RA

wird erst erkannt, wenn das Radar abfragt. Wegen der Radardrehung können bis zu 12 s vergehen, bis ein RA abgefragt werden kann. Die Event-Beispiele in Kapitel 9 zeigen, dass dies zu wenig ist, um den genauen Zeitpunkt des RA und der darauf folgenden Reaktion des Flugzeuges festzustellen.

Auch kann ein Mittelbereichsradar die Höhe des Flugzeuges nur alle 12 s ermitteln [Mensen]. Diese Totzeit ist zu groß, um die Reaktionen auf RAs zu beschreiben, die immer mit schnellen Änderungen der Steigraten (zweite Ableitung der Höhe) verbunden sind.

Da der RA-Downlink aber derzeit die einzige standardisierte Nachricht ist, in der ein Nicht-ACAS-ausgerüsteter Gegner genannt wird, ist seine Beobachtung auch für das ACAS-Monitoring von Bedeutung. Seit 2007 ist Mode S bei der DFS in betrieblicher Nutzung [DFS04]. Eine Einbindung des RA-Reports in das mithörende ACAS-Monitoring könnte somit einige ACAS-Event-Berichte vervollständigen.

3 Der ACAS-Monitoring-Ansatz

Im letzten Kapitel wurde aus dem ACAS-Standard eine ACAS-Monitoring-Idee entwickelt. In diesem Kapitel wird daraus ein Ansatz zur Beobachtung und Analyse der ACAS-Aktivitäten abgeleitet. Kernstück dieses Ansatzes ist die Beobachtung der Aussendungen von Flugzeugen im Mode S-Protokoll auf den Frequenzen 1030 MHz und 1090 MHz. Insbesondere müssen folgende Teile der ACAS-Kommunikation mitgehört werden:

- Die DF11-Squitter zeigen, welche Flugzeuge sich überhaupt im Empfangsbereich befinden.
- Die DF00-Air-Air-Surveillance-Antworten erzeugen ein Höhenprofil mit mehreren Updates pro Sekunde.
- DF17-ADS-B-Meldungen liefern zweimal pro Sekunde die Position der ADS-B-ausgerüsteten Flugzeuge.
- DF05-Radar-Antworten liefern die Zuordnung zwischen Squawk und Mode S-Adresse, um UF16-31 zu decodieren.
- Durch UF16-30-Resolution-Message wird die Identität des anderen Flugzeuges (Gegners) bekannt, sofern beide Flugzeuge mit ACAS ausgerüstet sind.
- Aus DF16-30-Coordination-Reply geht das Resolution Advisory (RA, Ausweichenweisung) des Gegners hervor, sofern beide Flugzeuge mit ACAS ausgerüstet sind.
- Aus UF16-31-RA-Broadcast kann das RA abgelesen werden, auch wenn der DF16-30-Coordination-Reply nicht empfangen wurde, z. B. weil der Gegner nicht mit ACAS ausgerüstet ist.

Daraus ergibt sich der in Bild 8 gezeigte erste grobe Ansatz für ein ACAS-Monitoring. Den Empfang übernimmt zunächst eine kombinierte 1030 MHz und 1090 MHz Mode S-Empfangsanlage. Später (Kapitel 8) wird das ACAS-Monitoring auf ein Netz von Empfangsstationen erweitert. Die Entwicklung der Empfangsanlagen ist nicht Teil dieser Arbeit, daher wird in Kapitel 4 lediglich ihre Anbindung an das ACAS-Monitoring gezeigt.

Die ACAS-Monitor-Software muss an die Empfangsanlagen angebunden sein, um aufgezeichnete oder in Echtzeit übertragene Nachrichten einzulesen. Die Nachrichten müssen anhand der AP- bzw. PI- Felder auf Übertragungsfehler überprüft werden. DF11-Squitter enthalten 000000_{hex} im PI-Feld. Diese DF11 können somit ohne Vorwissen geprüft werden und liefern im Nachrichtenkörper die Adressen im Luftraum befindlicher Flugzeuge. Die anderen Formate enthalten die Absenderadresse. Ist eine solche Absenderadresse schon einmal in einem DF11 genannt worden, kann sie als richtig angesehen werden.

Im nächsten Schritt werden die einzelnen Formate dekodiert. Die wichtigsten Formate sind in Bild 8 aufgeführt. DF11, DF17, DF00 und DF04 liefern Zustandsdaten die erst einmal unabhängig vom ACAS-Monitoring sind. Ihre Inhalte werden in ein Umgebungsmodell eingepflegt (Kapitel 5). Die eigentliche Rekonstruktion der ACAS-Dialoge kann nur anhand dieses Umgebungsmodells

erfolgen. Auf Grund von Übertragungsstörungen ist nicht zu erwarten alle zu einem ACAS-Eingriff gehörenden Nachrichten zu empfangen. Kapitel 6 zeigt, wie aus der unvollständigen Folge von mitgehörten Nachrichten der ACAS-Koordinationsdialog vollständig rekonstruiert wird.

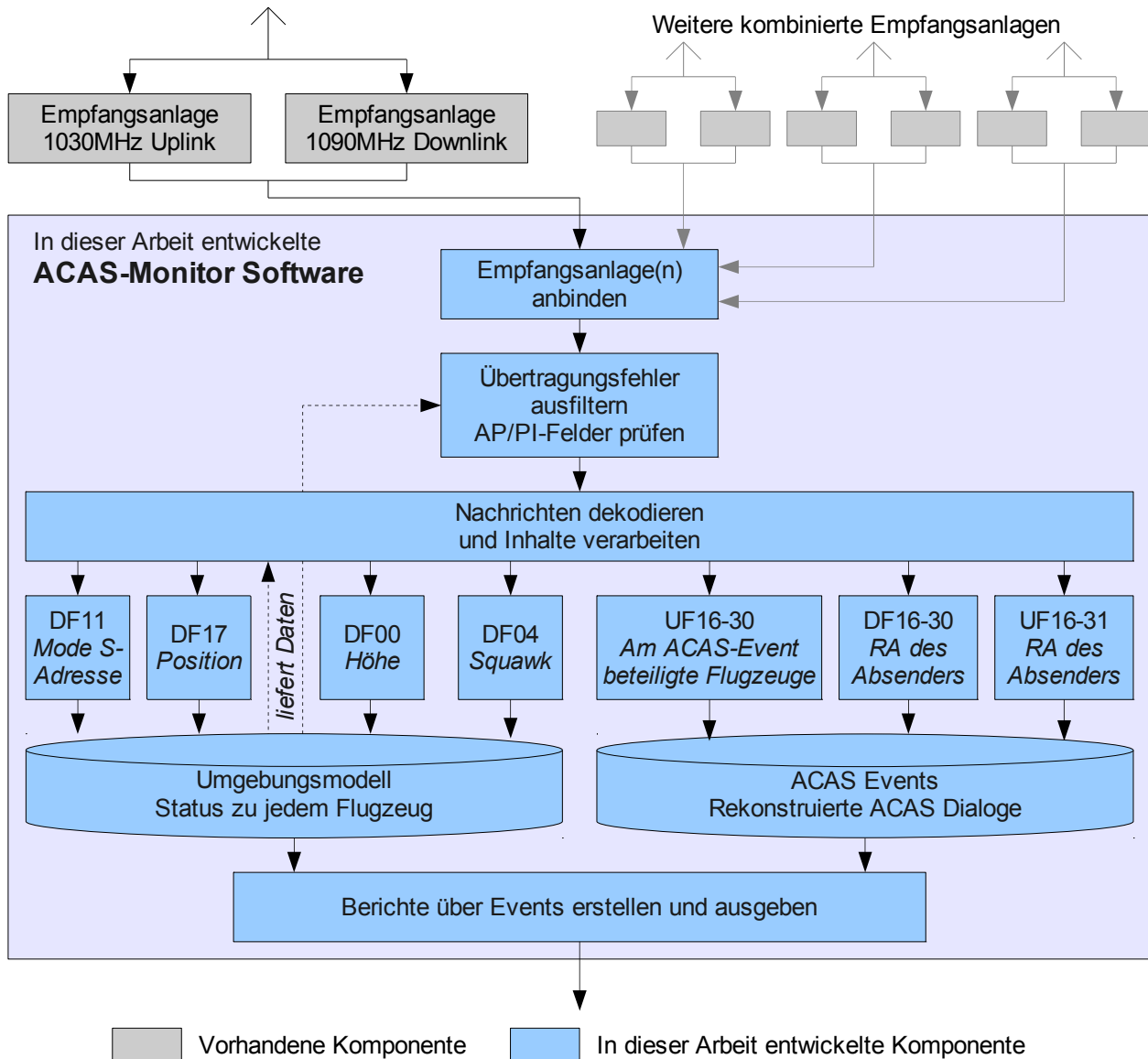


Bild 8: Ansatz für das ACAS-Monitoring

Im letzten Schritt werden die ACAS-Koordinationsdialoge mit Bahndaten der Flugzeuge zusammengebracht und zu Event-Berichten verarbeitet. Damit ist ein ACAS-Monitoring mit mithörenden Bodenstationen möglich, das die Funktion aller ACAS-Geräte im Luftraum beobachtet, analysiert und automatisch Berichte erstellt. In diesen Berichten kann die vertikale Situation vor dem ACAS-Eingriff, die Anweisung des ACAS und die vertikale Situation nach dem ACAS-Eingriff beschrieben werden. Mit ADS-B kann zusätzlich auch noch die horizontale Situation beschrieben werden.

Damit wird nun ein ACAS-Monitor entwickelt, der automatisch die Verkehrssituation vor dem ACAS-Eingriff, den Koordinations-Dialog, das RA selbst und die Reaktion der beteiligten Flugzeuge objektiv dokumentiert. Die Aufgabe besteht nun darin, diesen Ansatz in ein System aus Hard- und Software umzusetzen. Im folgenden Kapitel 4 wird daher zunächst die Anbindung der verwendeten Empfangs-Anlagen beschrieben.

4 Empfang von Mode S-Nachrichten – Anbindung von Mode S-Empfangsstationen

Bevor das ACAS-Monitoring Mode S-Nachrichten auswerten kann, müssen sie zuerst empfangen und digitalisiert werden. Die dazu nötigen Empfangsanlagen, ihre Anbindung an die in dieser Arbeit entwickelte Auswertesoftware und die nötigen Datenformate werden in diesem Kapitel beschrieben.

Die Empfangsanlage wird an eine Antenne angeschlossen. Sie empfängt die Signale aus den Mode S-Kanälen 1030 MHz und 1090 MHz und gibt sie an einen Computer als Folge von Mode S-Nachrichten aus. Im Projekt standen zwei Typen von Empfangsanlagen zur Verfügung: Eine Eigenentwicklung der TU Braunschweig und 1030 MHz bzw. 1090MHz Empfangsanlagen der Firma Thales.

In dieser Arbeit steht die Entwicklung von Empfangsanlagen selbst **nicht** im Mittelpunkt, dennoch ist ACAS-Monitoring auf leistungsfähige Empfangsanlagen angewiesen. Erkenntnisse aus dem ACAS-Monitoring – speziell zur Kanalbelastung – können bei der Weiterentwicklung von Empfangsanlagen helfen.

4.1 Experimentelle Empfangsanlage der TU Braunschweig

Am Institut für Eisenbahnwesen und Verkehrssicherung (IfEV) wurden und werden von Prof. Dr.-Ing. Peter Form und Dr. Jochen Bredemeyer mehrere Mode S-Empfangsanlagen entwickelt [Bredemeyer03]. Die zweite gebaute Empfangsanlage wurde zur Daueraufzeichnung im SAT-T¹⁷-Projekt seit 20. April 2007 benutzt. Diese Mode S-Empfangsanlage kombiniert Up- und Downlink-Empfänger und war lange Zeit die einzige Anlage zum Empfang von Uplink-Nachrichten, die dem Projekt zur Verfügung stand. Hier soll nun dargestellt werden, wie die Empfangsanlage an den ACAS-Monitor-Prozess angebunden wurde.

Bild 9 zeigt den schematischen Aufbau der gesamten ACAS-Monitor-Anlage, bestehend aus der experimentellen Empfangsanlage und dem Analyserechner. Von einer Antenne werden Mode S-Signale zusammen mit generierten Testsignalen (Abschnitt 4.3.1) an das Empfängerpaar für 1030 MHz und 1090 MHz übergeben. Die Korrelation, Phasenerkennung und Demodulation ist in einen FPGA implementiert¹⁸. Die digitalisierten Mode S-Nachrichten werden per USB an den Analyserechner ausgegeben. Dort werden die Daten vom USB-Treiber fx2driver.ko, der Aufzeichnungssoftware `ssr_corr` und der Konvertierungssoftware `ssr_extract` weiter verarbeitet und schließlich dem eigentlichen ACAS-Monitoring zugeführt. Der Analyserechner arbeitet mit dem Open Source Betriebssystem Linux [Linux].

¹⁷ Das SAT-T-Projekt (Safety Through Transparency) war ein Teilprojekt des Luftfahrtforschungsprogramms III des Bundesministeriums für Wirtschaft und Technologie. Das Ziel war, die Grundlagen des ACAS-Monitorings zu legen. Partner waren die Deutsche Flugsicherung DFS, Thales GmbH Stuttgart und das Institut für Eisenbahnwesen und Verkehrssicherung der TU Braunschweig.

¹⁸ Ein FPGA (Field Programmable Gate Array) ist ein programmierbarer integrierter Schaltkreis. In FPGAs können durch spezifische Konfiguration interner Strukturen die verschiedenartigsten digitalen Schaltungen gebildet werden.

Auf dem Analyserechner kontrolliert ein im Rahmen dieser Arbeit entwickeltes Steuerprogramm den Ablauf der Aufzeichnung und Auswertung. Es überwacht den FPGA, die USB-Anbindung, fx2driver.ko und ssr_corr. Es startet nach 24 Stunden eine neue Aufzeichnungsdatei. Und es lässt parallel zur neuen Aufzeichnung die des Vortages mit ssr_extract und dem ACAS-Monitor auswerten.

Programm zur Datenaufzeichnung – ssr_corr

Das Programm ssr_corr liest Mode S-Daten vom Kernel Modul fx2driver.ko (dem USB-Treiber) und schreibt sie unverändert als Binärdatei auf die Festplatte. Ssr_corr sowie fx2driver.ko wurde von Dr. Jochen Bredemeyer entwickelt.

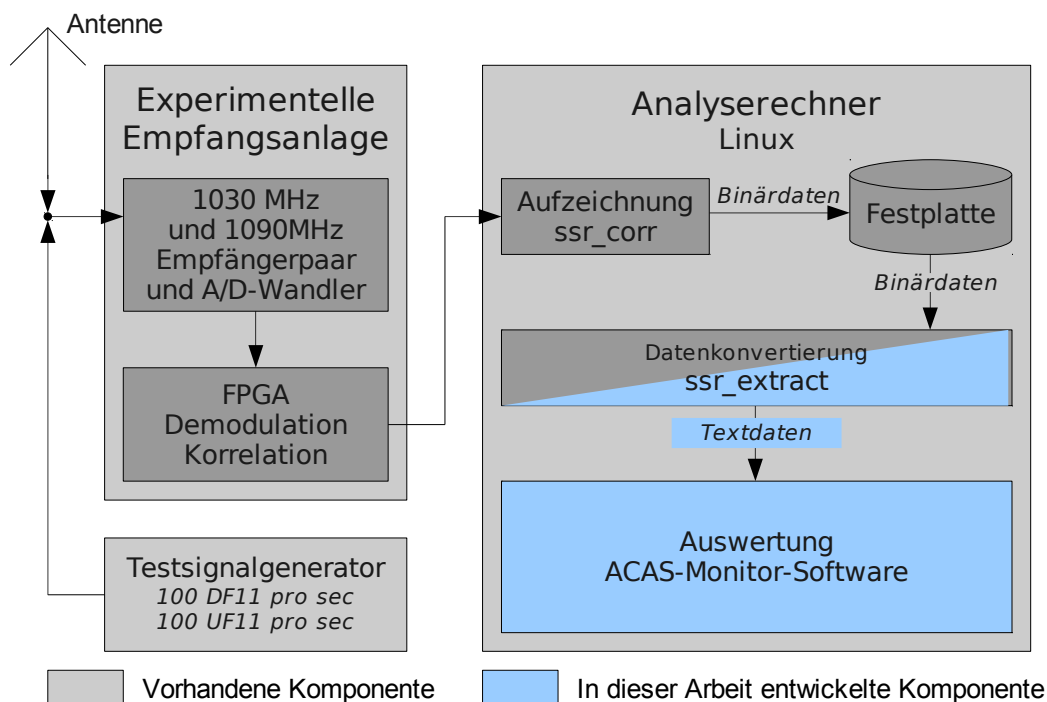


Bild 9: Anbindung der experimentellen Empfangsanlage der TU Braunschweig

Ssr_extract

Das binäre Aufzeichnungsformat der experimentellen Empfangsanlage hat den Nachteil, dass es nicht ohne eigens dafür entwickelte Software auswertbar ist und dass es sich mit jeder Version der Empfängerfirmware ändern kann – was auch wirklich eingetreten ist. Zur Anbindung und Entwicklung der ACAS-Monitor-Software musste im Rahmen dieser Arbeit eine einfacher zu lesende und von dem Empfänger abstrahierte Darstellung der Mode S-Daten gefunden werden. Es wurde ein Textformat gewählt, dass sich einfach mit awk [AWK], Python [Python] oder einem Texteditor analysieren lässt. Die Umwandlung der Binärdaten in das Rohdaten-Textformat übernimmt der Konverter ssr_extract. Die von ssr_extract ausgegebenen Daten können direkt von der entwickelten ACAS-Monitor-Software verarbeitet werden.

4.1.1 Aufbau des Rohdaten-Textformats der TU Braunschweig

Listing 1 zeigt einen Ausschnitt aus einer Aufzeichnung im Rohdaten-Textformat.

```

DF11 Zeit: 30506576.24 ms Nr: 18432 Lev: -65 dBm AP: 000007 Data: 5D4B161652F079
DF00 Zeit: 30506576.50 ms Nr: 18433 Lev: -69 dBm AP: 400C50 Data: 02E1991058EB1D
DF11 Zeit: 30506577.89 ms Nr: 18434 Lev: -65 dBm AP: 000003 Data: 5D4B161652F07D
DF17 Zeit: 30506582.85 ms Nr: 18438 Lev: -72 dBm AP: 1C1B8A Data: 8D4A81BB68B50300EBAFCB7F3D36
DF11 Zeit: 30506583.10 ms Nr: 18437 Lev: -68 dBm AP: 000000 Data: 5D3C666AC1E39B

```

Listing 1: Ausschnitt im Rohdaten-Textformat der TU Braunschweig

Der Aufbau ist weitgehend selbsterklärend. Die Felder enthalten (von links nach rechts):

1. Nummer des Mode S-Formats mit Kennzeichnung für Uplink (UF) und Downlink (DF).
2. Zeitstempel, zu dem die Nachricht von der Empfangsanlage empfangen wurde in Millisekunden seit Tagesanfang in der Zeitzone des Computers, mit dem die Daten aufgezeichnet wurden.
3. Laufende Nummer des Paketes, diese Nummer wird von der Empfangsanlage generiert, Uplink und Downlink Nachrichten werden unabhängig voneinander gezählt.
4. Die Empfangsleistung in dBm am Eingang des Empfängers.
5. Das AP-Feld (bei DF11, DF17, DF18 bezeichnet der Standard [ICAO Annex10v4] das AP-Feld als PI-Feld).
6. Der Inhalt der Nachrichten, exakt wie sie empfangen wurden, in hexadezimaler Schreibweise. Die Nachrichten können kurz oder lang sein.

4.2 Thales Empfangsanlage

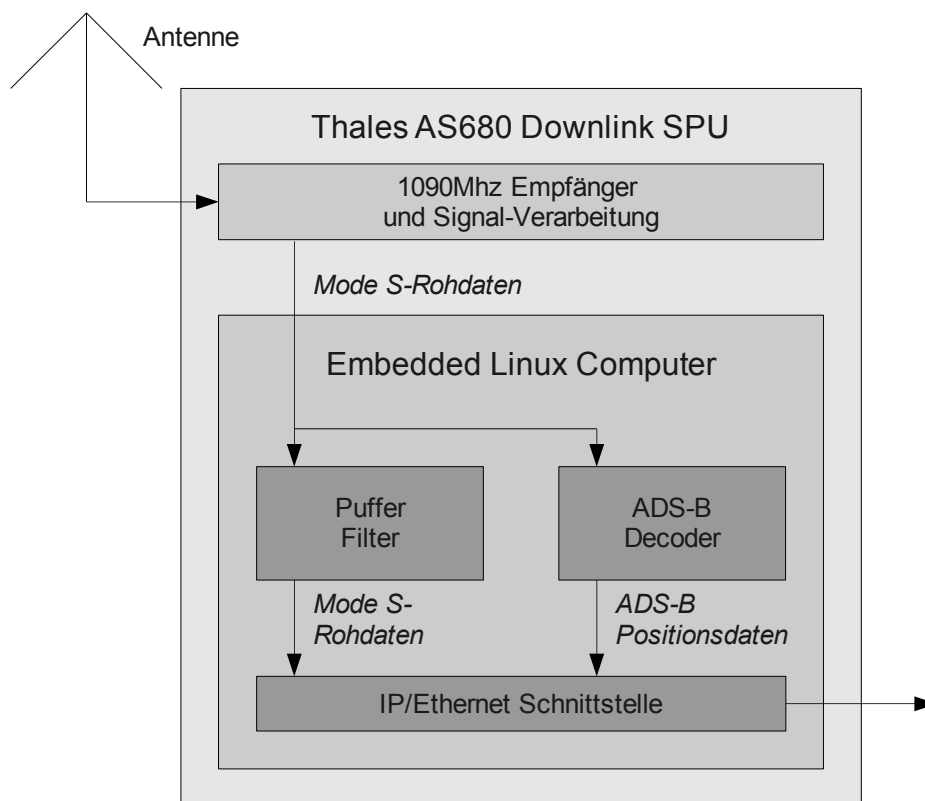


Bild 10: Schematischer Aufbau der Thales AS680 Downlink Empfangsanlage

Die Thales GmbH vertreibt Bodenstationen für den Mode S-Downlink unter der Bezeichnung AS680, die Uplink Empfangsanlagen haben keine eigene Bezeichnung. Die Empfangsanlage der AS680 bildet die sog. SPU (Signal Processing Unit). Es gibt Varianten der SPU für den Uplink (1030 MHz) und für den Downlink (1090 MHz). Zu Beginn des Projektes stand nur die Downlink-Variante (Bild 10) zur Verfügung, Thales konnte die Entwicklung der Uplink-Variante 2008 abschließen. Beide Empfangsanlagen unterscheiden sich nur im Hochfrequenz-Teil, die digitalen Verarbeitungseinheiten und die Softwareanbindung ist weitgehend identisch. Die SPU ist als ein 19"-Gerät ausgeführt (133 x 482 x 250 mm), je eine Uplink-SPU und eine Downlink-SPU werden den Kern einer Bodenstation des späteren ACAS-Monitor-Netzes in Kapitel 8 bilden.

Die Mode S-Empfangsdaten werden vom 1090MHz-Empfänger empfangen und danach digitalisiert. Ein embedded Linux Computer puffert und filtert die Rohdaten, im Downlink Empfänger werden noch die ADS-B-Positionsdaten decodiert. Alle Daten werden über eine IP/Ethernetschnittstelle in UDP¹⁹ [UDP] Paketen ausgegeben. ADS-B-Positionsdaten werden im Asterix-CAT-21-Format [ASX CAT21] ausgegeben. Die Rohdaten werden in einem ähnlich aufgebauten Format ausgegeben (siehe 4.2.1).

Die Thales SPU's und die dazugehörigen Tools werden im Manual der Bodenstation [Thales06] von Thales erläutert. Hier soll nur die Anbindung des ACAS-Monitors besprochen werden.

4.2.1 Thales Rohdaten-Binärformat

Die Thales Uplink- und Downlink-SPU geben die empfangenen Mode S-Nachrichten per UDP¹⁹ im Thales Rohdaten-Binärformat (.thbin) aus. Das Downlink-Rohdatenformat wird in [Thales05] von Thales beschrieben, das Uplink-Rohdatenformat ist fast identisch, wurde von Thales aber nicht dokumentiert und musste in dieser Arbeit durch Reverse Engineering²⁰ offen gelegt werden.

Jede Mode S-Nachricht wird in ein Rohdatenpaket wie in Bild 11 (oben) gepackt. Mehrere Rohdatenpakete können in ein UDP-Paket gepackt werden.

Thales Rohdatenformat:

ID	FSPEC	SAC	SIC	MSG	TOD	TOA	NCB	SA
:8	:8	:8	:8	:56 (kurz) :112(lang)	:24	:40	:8	:8

Erweitertes Thales Rohdatenformat:

SIZE	ID	FSPEC	SAC	SIC	MSG	TOD	TOA	NCB	SA	DATE
:8	:8	:8	:8	:8	:56 (kurz) :112(lang)	:24	:40	:8	:8	:16

Bild 11: Thales Rohdatenformat und Erweiterung der TU Braunschweig

Die Thales Rohdatenpakete enthalten:

- ID Rohdatennachrichtentyp, 01_{hex} für Downlink-Nachrichten, 03_{hex} für Uplink
- FSPEC Zeigt an, ob MSG lang oder kurz ist und welche der optionalen Felder TOD, TOA, NCB und SA in diesem Rohdatenpaket vorhanden sind

¹⁹ UDP: User Datagram Protocol ist ein minimales, verbindungsloses Netzwerkprotokoll, das zur Transportschicht der Internetprotokollfamilie gehört.

²⁰ Beim Reverse Engineering werden Systemdokumentationen dadurch erstellt, dass die fertigen Systeme in Aufbau und Funktion untersucht werden.

- SAC System Area Code, Kennung des Landes, in dem die Anlage aufgestellt ist
- SIC Identifikation des Empfängers
- MSG Eigentlichen Datenbits der kurzen oder langen Mode S-Nachricht
- TOD Empfangszeitpunkt der Mode S-Nachricht, datiert nach der Systemuhr des Embedded Linux Computers in 1/128 s seit Mitternacht
- TOA Empfangszeitpunkt der Mode S-Nachricht, datiert mit einem Zähler in der Signalverarbeitungseinheit mit einer Auflösung von 7,8125 ns. Mit diesem Zähler können Nachrichten hoch-genau datiert werden, er läuft allerdings ca. alle zwei Stunden über. Die Auflösung von TOD ist für das ACAS-Monitoring ausreichend, daher wird TOA in dieser Arbeit nicht weiter verwendet.
- NCB Anzahl der nicht sicher demodulierten Bits in der Mode S-Nachricht. Ein ideal demoduliertes Signal hat NCB = 00
- SA Empfangsleistung in -1 dBm am Empfängereingang, bei Uplink-Nachrichten. 1/256 Anteile von 2V am Empfängereingang bei Downlink-Nachrichten.

Eine kurze Mode S-Nachricht ist damit 21 Byte lang, eine lange 28 Byte. Dateien mit Aufzeichnungen im Thales Binärformat wurden in diesem Projekt mit der Dateinamensendung .thxbn gekennzeichnet.

4.2.2 TU Braunschweig Erweiterung zum Thales Rohdaten-Binärformat

Das binäre Rohdatenformat der Thales-SPUs speichert nicht das Datum einer empfangenen Nachricht. Außerdem muss eine Software, die Nachrichten nur blockweise weiterleiten soll, bei jeder Nachricht das FSPEC-Feld decodieren, um die Nachrichtenlänge zu berechnen. Wird das Format erweitert, müssen alle Programme angepasst werden, da sie die neuen Blocklängen nicht berechnen können. Wegen dieser Missstände, wurde im Rahmen der Arbeit ein erweitertes Rohdatenformat definiert (.thxbn, Bild 11 unterer Teil). Es sind folgende Daten hinzugekommen:

- SIZE Länge der folgenden Nachricht in Bytes (exklusive dem SIZE Byte)
- ID Rohdatennachrichtentyp
Thales Rohdaten: 01_{hex} für Downlink-Nachrichten, 03_{hex} für Uplink
Erweitertes Rohdatenformat 81_{hex} für Downlink, 83_{hex} für Uplink
- FSPEC Bit 7 zeigt an, das ein DATE-Feld vorhanden ist
- DATE Datum, als Anzahl der Tage seit 01. Januar 2000, einschließlich Schalttage

Das erweiterte binäre Rohdatenformat ist somit 3 Byte länger, was 25 bzw. 31 Bytes pro Nachricht ergibt. Dateien mit Aufzeichnungen im Erweiterten Binärformat wurden in diesem Projekt mit der Dateinamensendung .thxbn gekennzeichnet. Das Erweiterte Binärdatenformat wird genutzt, um Mode S-Daten zu puffern, gepufferte Daten via Internet zu übertragen und um die Mode S-Aufzeichnungen zu archivieren.

4.2.3 Thales Raw Data Logging Format

Zu den oben beschriebenen Binärdatenformaten gibt es die zugehörigen Textdarstellungen, von Thales als Logging Format bezeichnet, Listing 2 zeigt einen Auszug aus solchen Daten. Im Thales Logging Format gibt es kein Feld, das angibt, ob die Daten aus dem Uplink oder dem Downlink stammen. Das Format und die dazugehörige Software wurden definiert, bevor es von Thales Uplink-Empfangsanlagen gab.

```
090 006 52379.552 52479.117 6118.612696969 20 3C6590 A0001217000000324800003C6590 01 32
090 006 52379.552 52479.117 6118.617965531 11 4A819C 5D4A819C00000100000000000000 00 3D
090 006 52379.552 52479.125 6118.625527164 11 4A819C 5D4A819C00000100000000000000 00 3D
090 006 52379.552 52479.125 6118.625796094 04 484451 200010BA48445100000000000000 00 47
```

Listing 2: Ausschnitt aus Aufzeichnung im Thales Logging Format

Die Daten-Felder (Spalten) enthalten folgende Daten aus dem binären Rohdatenformat (Abschnitt 4.2.1):

SAC	SIC	TOR	TOD	TOA	Format	AP	MSG	NCB	SA
-----	-----	-----	-----	-----	--------	----	-----	-----	----

Wobei TOR der Zeitpunkt ist, zu dem die Nachricht von der Thales-Aufzeichnungssoftware am Computer empfangen wurde (Sekunden seit Mitternacht auf dem Computer, auf dem die Aufzeichnungssoftware läuft, Genauigkeit von 3 Dezimalstellen).

Um Daten im Erweiterten Binärformat als Text darzustellen, wurden im Rahmen dieser Arbeit die entsprechenden Spalten angehängt, siehe Listing 3.

Spalten:

1	2	3	4	5	6	7	8	9	10	11	12
107	015	-----	64530.219	3559.211380195	05	4D00C5	2800199F4D00C50000000000000000	00	4B	DL	20090215
107	015	-----	64530.234	3559.212738297	00	4841DA	02E617954841DA000000000000000	02	36	DL	20090215
107	015	-----	64530.234	3559.213936336	11	4B161E	5D4B161E0000020000000000000000	00	4F	DL	20090215
107	014	-----	64530.234	4496.034812039	16	FFFFFF	80800000320000000400C4AFFFFFFF	00	42	UL	20090215
107	016	-----	64530.234	4496.093692938	16	FFFFFF	808000003200000004CA4F7FFFFFFF	02	4B	UL	20090215
107	013	-----	64530.203	3559.736187469	11	F0001C	5BF0001C0000000000000000000000	00	3D	DL	20090215

Listing 3: Ausschnitt im Logging Format mit Erweiterungen der TU Braunschweig

Spalte 11 kennzeichnet das Format mit UL und DL, in Spalte 12 wird das Empfangsdatum genannt. Die hier gezeigten Daten wurden mit dem Programm Thxbin2txt (siehe unten) erstellt, Spalte 3 (TOR, Empfangszeitpunkt durch den aufzeichnenden Computer) wird von diesem Programm nicht ausgegeben, sondern mit „-----“ gefüllt.

4.2.4 Software zur Verwaltung der einfachen und erweiterten Datenformate der Thales Empfangsanlagen

Um mit den Mode S-Datenformaten umgehen zu können, wurden die nötigen Konvertierungsprogramme im Rahmen dieser Arbeit entwickelt.

Thxbin2txt

Thxbin2txt konvertiert binäre Rohdaten-Dateien in das Thales Raw Data Logging Format. Das in dieser Arbeit entwickelte Programm kann das einfache und das erweiterte binäre Rohdatenformat (.thbin bzw. .thxbin) lesen. Da Mode S-Daten eine geringe Shannon-Entropie²¹ haben, ist es sinn-

²¹ Shannon-Entropie [Shannon]: Maß der Informationsdichte, vereinfacht dargestellt: Kommen in Daten häufig die gleichen Muster

voll, Mode S-Dateien immer zu komprimieren. Im Projekt wird hauptsächlich Bzip2 [Bzip2] eingesetzt. Thxbin2txt erkennt automatisch, ob es sich um komprimierte Daten handelt und entpackt diese beim Konvertieren.

Mit dem Programm können binäre Aufzeichnungen leicht entpackt, dargestellt und in weiteren experimentellen Programmen verarbeitet werden. So wurde auch Listing 3 mit Thxbin2txt erstellt.

Thudp2txt

Dieses in dieser Arbeit einwickelte Programm nimmt binäre Mode S-Daten von den Thales SPUs an und gibt sie als erweiterten Text aus. Mit Thudp2txt kann leicht die Funktion einer Empfangsanlage geprüft werden, ohne dass erst Daten aufgezeichnet werden müssen.

Thudp2binfile

Auch das Programm Thudp2binfile musste im Rahmen dieser Arbeit neu entwickelt werden. Mit Thudp2binfile werden Binärdaten, die von Empfangsanlagen per UDP über das Netz gesendet wurden, aufgezeichnet. Die Aufzeichnungen werden automatisch beim Schreiben der Dateien komprimiert, in mehrere Dateien aufgeteilt und mit Aufnahmezeitpunkt benannt. Wird das Ziellaufwerk voll, so wird automatisch auf ein alternatives Laufwerk weiter gespeichert.

Dieses Programm erfüllt damit die Aufgaben des kompletten Aufnahmeprozesses, der Mode S-Daten in handhabbaren Dateien aufzeichnet. Dies geschieht am besten in einer Ramdisk²², aus dem die Aufnahmen von einem anderen Prozess regelmäßig entnommen und an die Zentrale geschickt werden. Wird die Ramdisk voll, so kann auf die normale Festplatte zurückgegriffen werden.

4.2.5 ADS-B-Positionsdaten – Asterix CAT-21

In den Thales AS680 Downlink-SPU ist ein ADS-B-Decoder integriert, dessen Ausgabe wird über UDP als Asterix CAT-21-Daten an eine beliebige IP-Adresse geleitet. Dies geschieht parallel zur Übertragung der Mode S-Rohdaten über UDP.

Recordasx

Die Asterix CAT-21-Daten können mit den Thales Record Asterix Data Tool (recordasx) aufgezeichnet oder in Textform ausgegeben werden. Listing 4 zeigt ein Beispiel für eine Text-Ausgabe, wobei jede Zeile in der Ausgabe umgebrochen werden musste und hier als ein Absatz dargestellt ist. Jede Zeile (hier als Absatz dargestellt) repräsentiert eine Asterix-CAT-21 Nachricht. Die Felder der Nachricht sind durch „|“ getrennt. Eine Nachricht enthält unter anderem die Position des Flugzeuges, die Mode S-Adresse und die Druckhöhe. Eine vollständige Dokumentation der Felder enthält das Manual der Bodenstation [Thales06].

vor, so haben sie eine geringe Entropie und lassen sich gut komprimieren. Daten im binären Thales-Rohdatenformat lassen sich mit Bzip2 im Mittel um den Faktor 2,5 verkleinern, bereits komprimierte Daten dagegen haben eine hohe Entropie und lassen sich mit Bzip2 nicht weiter komprimieren.

22 Ramdisk ist ein virtueller und nicht persistenter Datenträger im Arbeitsspeicher eines Computers.

```

90|6|0|0|0|0|0|0|0|0|0|1|1|54313.500|52.537114620|8.574142456|3C48AC|-|0|0|0|0|0|1|0|0|0|
-|220.00|-|-|-|-|0.00|-|0.116638|208.7622|-|-|DLH083  |0|-|0|-|-|-|-|-|-|-|-|-|-|
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
90|6|0|0|0|0|0|0|0|0|0|1|1|54313.656|52.010929585|10.867452621|3C1198|30400.00|0|0|0|6|0|
1|0|0|0|-|299.75|-|-|-|-|-|0.00|0.097900|353.3203|-|-|DLH9JX  |2|-|0|-|-|-|-|-|-|-|-|-|-|
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|
90|6|0|0|0|0|0|0|0|0|0|1|1|54313.750|52.234818935|12.451972961|400492|32850.00|0|0|0|0|0|
1|0|0|0|-|360.00|-|-|-|-|-|62.50|-|0.104858|277.7673|-|-|LS596  |0|-|0|-|-|-|-|-|-|-|-|-|-|
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|

```

Listing 4: Ausschnitt aus Asterix CAT-21 Daten in Textform

Replayasx

Mit dem Thales Replay Asterix Data Tool (replayasx) können binäre oder textuelle Asterix-Aufzeichnungen als Asterix CAT-21-Daten an Netzwerkadressen gesendet werden, wo z. B. ein ADSB-Display (s. u.) die Daten entgegennimmt. Replayasx kann z. B. folgendermaßen aufgerufen werden:

```
$ cat meine_datens.asx | replayasx -d 134.169.150.140 -p 9912
```

Die ACAS-Monitor-Software, die in den folgenden Kapiteln entwickelt wird, enthält auch einen im Projekt entwickelten ADS-B-Decoder, der Textdaten für Replayasx erzeugen kann.

ADSB-Display

Der Thales Technical Situation Display (ADSB-Display) dient dazu, Asterix CAT-21-Daten darzustellen. Bild 14 auf Seite 38 zeigt die Spuren, dargestellt mit dem ADSB-Display.

4.3 Empfangsstatistiken

Der zu entwickelnde ACAS-Monitor soll Mode S-Nachrichten von Flugzeugen analysieren. Dabei können bei weitem nicht alle Mode S-Nachrichten, die ein Flugzeug sendet, empfangen werden: Nachrichten können von anderen, gleichzeitig ausgesendeten Mode A/C oder Mode S-Nachrichten gestört werden. Fremde Störquellen, wie nahe Mobilfunksender, können den Empfang auf den Mode S-Kanälen stören. Komponenten der Empfangsanlagen oder Antennen können ihre Eigenschaften verändern und ganz oder teilweise ausfallen. Die aufzeichnende Software kann überlastet sein oder abstürzen.

Bei der Entwicklung und beim Betrieb eines ACAS-Monitors müssen also genaue Kenntnisse vom realen Mode S-Funkfeld vorliegen. Zu diesem Zweck erstellt die ACAS-Monitor-Software täglich automatisch Statistiken, die die aktuelle Situation im Funkfeld beschreiben. Bereits vor der Entwicklung der ACAS-Monitor-Software wurden Funkfeldstatistiken erstellt, um festzustellen, welche und wie viele Daten zu erwarten sind, und ob die Empfangsanlagen überhaupt kontinuierliche glaubhafte Daten liefern. Da die ACAS-Monitor-Software alle Mode S-Nachrichten analysiert, kann sie auch leicht als „Mode S-Statistik-Software“ angepasst werden, das ist im Laufe der Entwicklung immer wieder geschehen. Die resultierenden Statistiken dienten zur Verbesserung und Kontrolle der Empfangsanlagen (Abschnitt 8.3) und halfen bei Designentscheidungen in der Entwicklung des ACAS-Monitors (Abschnitt 8.1). Hier sollen nur die wichtigsten Statistiken gezeigt werden.

4.3.1 Testnachrichten

Es ist einfach zu zählen, wie viele Nachrichten empfangen werden. Genauso interessant ist aber auch die Anzahl der gesendeten, aber nicht empfangenen Nachrichten, denn daraus ergibt sich die Empfangswahrscheinlichkeit. Um Aussagen über die Empfangswahrscheinlichkeit von Mode S-Nachrichten machen zu können, werden jede Sekunde 100 Testnachrichten mit kleiner Leistung (-80 dBm) in die Antennenleitung eingespeist. Die Testnachrichten sind UF11-All-Calls bzw. DF11-Squitter. Damit wird die Übertragung von einem entfernten Flugzeug simuliert ohne die Funkfeldlast selbst zu erhöhen. Diese Nachrichten werden von der ACAS-Monitor-Software herausgefiltert und es wird gezählt, wie viele Testtelegramme pro Sekunde ungestört empfangen wurden.

4.3.2 Funkfeldstatistik eines Tages

Bild 12 und Bild 13 zeigen automatisch erstellte Funkfeldstatistiken, die während der Daueraufzeichnung (Kapitel 9) entstanden. Als Beispiel wurde ein Tag gewählt, an dem alle Empfangsanlagen der TU Braunschweig arbeiteten – willkürlich fiel die Wahl auf den 05. September 2008. Empfangsstatistiken anderer Tage sind in [Gottst07] und [Gottst08] zu finden.

Bild 12 zeigt die Funkfeldstatistik für die experimentelle Empfangsanlage der TU Braunschweig mit der Antenne auf dem Dach des Institutes. Mit diesem System wurde der erste Teil der Daueraufzeichnung von April 2007 bis April 2008 bestritten. Ab Mai 2008 wurde diese einzelne Station in Braunschweig schrittweise durch ein deutschlandweites Netz von Empfangsstationen ersetzt. Bild 13 zeigt die Funkfeldstatistik des Netzes mit Stationen in Braunschweig, Frankfurt, Oberpfaffenhofen und Korntal, wobei in Braunschweig eine Doppelstation mit Sektorantennen nach Nordwesten und Südosten aufgebaut ist.

Die Statistiken enthalten Diagramme über den Verlauf eines Tages für jeden Empfänger und jeden Kanal. Das obere linke Diagramm in Bild 12 zeigt als rote Kurve (linke Ordinate) die Anzahl der insgesamt von der experimentellen TU-Empfangsanlage empfangenen Downlink-Nachrichten pro Sekunde. Zwischen 07.00 und 22.00 Uhr sind das circa 1500 Nachrichten pro Sekunde. Die darunter liegende grüne Kurve zeigt die Rate²³, der durch CRC-Prüfung (Abschnitt 2.4) als korrekt anerkannten Nachrichten. Das sind zwischen 07.00 und 18.00 Uhr rund 500 Nachrichten pro Sekunde. Schon hier wird deutlich, dass ein großer Teil der Nachrichten nicht korrekt empfangen wurde.

Die blaue Kurve zeigt auf der rechten Ordinate, wie viele der 100 Testnachrichten pro Sekunde in allen Bits fehlerfrei empfangen wurden – das sind zwischen 07.00 und 22.00 Uhr nur ca. 40%. Daraus folgt, dass zwischen 07.00 und 22.00 Uhr auch nur ca. 40% der realen mit -80dBm empfangenen Nachrichten nicht durch andere Mode S-Nachrichten gestört werden. Die Statistik der Testnachrichten zeigt auch, dass die Empfangswahrscheinlichkeit von der Kanalbelastung abhängt: Nachts, wenn wenige Flugzeuge unterwegs sind und somit wenig gesendet wird, ist die Empfangswahrscheinlichkeit fast 100%. Die Testsignale sind in den roten und grünen Kurven nicht mehr enthalten.

Oben rechts in Bild 12 ist die Uplink-Statistik aufgezeichnet. Die rote Kurve zeigt die Rate aller empfangenen Nachrichten (linke Ordinate), das sind in den Tagesstunden circa 500 Nachrichten pro Sekunde. Darunter ist grün die Rate der CRC-korrekten Nachrichten abgetragen, zirka 400 Nach-

²³ Die Rate bezeichnet die Anzahl von Ereignissen (hier Nachrichten) pro Sekunde

richten pro Sekunde. Die Empfangswahrscheinlichkeit (blaue Kurve) im Uplink bricht am Tag deutlich weniger ein als im Downlink. Das liegt zum einen an der geringeren Funkfeldlast im Uplink. Zum anderen wird im Uplink nur die Präambel der Nachrichten amplitudenmoduliert, für die Nachricht selbst wird differentielle Phasenumtastung verwendet, die robuster gegen Störungen ist.

Das letzte Diagramm zeigt, dass sich zwischen 07.00 und 22.00 Uhr ca. 250 Mode S-ausgerüstete Verkehrsflugzeuge (rot) gleichzeitig im Empfangsbereich der Experimentellen Empfangsanlage der TU Braunschweig befinden. Davon sind auch fast alle mit ACAS ausgestattet (grün), 60% senden außerdem ADS-B (blau). Die Kurven zittern im Zehn-Minuten-Takt, weil die Liste aller bekannten Flugzeugadressen von der ACAS-Monitor-Software jede 10 min bereinigt wird.

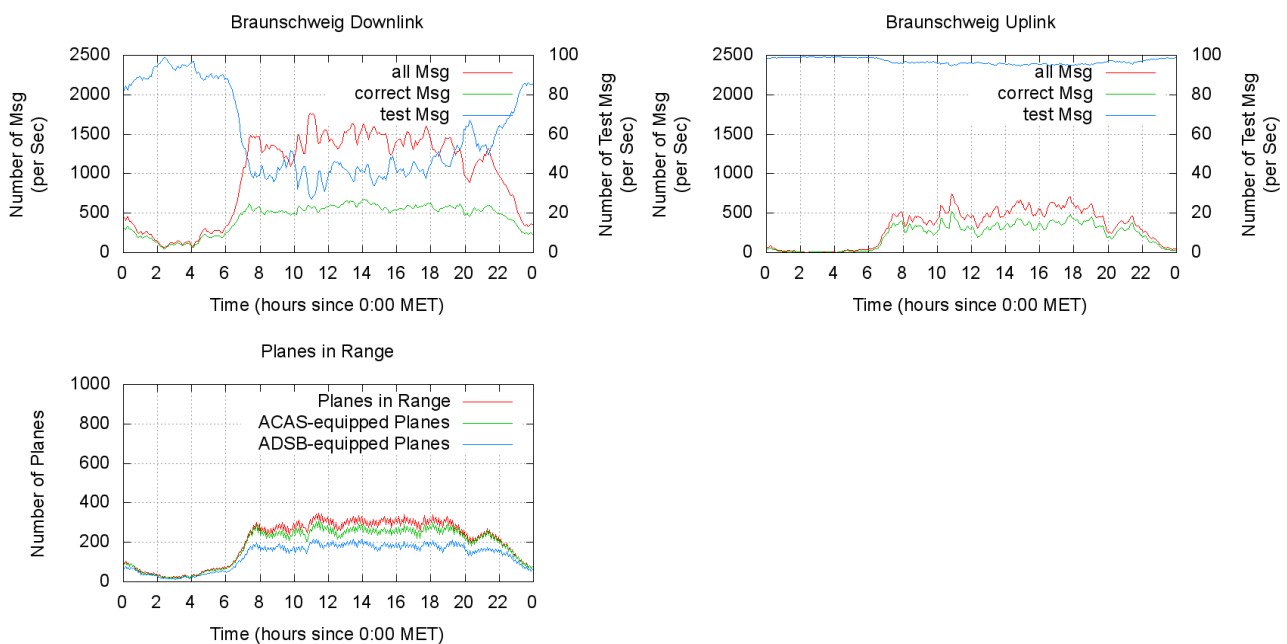


Bild 12: Funkfeldstatistik der Experimentellen Empfangsanlage der TU Braunschweig

Eine ähnliche Statistik wird jeden Tag für die Empfänger des ACAS-Monitor-Netzes erstellt (Bild 13). Zum ACAS-Monitor-Netz gehören heute (Ende 2009) fünf Downlink- und fünf Uplink-Empfangsanlagen. Die Empfangsanlagen sind alle Thales SPUs, wobei in Braunschweig eine Doppelstation mit Sektorantennen nach Nordwesten und Südosten aufgebaut ist.

Bild 13 ist mit der gleichen Software erstellt, wie Bild 12. Nur die Empfangsanlagen in Braunschweig werden mit Testsignalen beaufschlagt, die als blaue Kurven im Diagramm eingetragen sind. Die Empfangsanlagen sind über Internet an die Zentrale im Institut angebunden. Um Übertragungsrate zu sparen, werden bereits an der Empfangsanlage die Nachrichtenformate ausgefiltert, die für das ACAS-Monitoring nicht relevant sind, das sind UF00, UF04, UF05, UF11, UF20 und UF21 (alle Uplink Formate außer UF16). In Braunschweig werden die UF11 nicht ausgefiltert, da die Testsignale das Format UF11 haben. Aus der Filterung resultieren die geringen Raten im Uplink und die Anpassung der Ordinatenkalierung in den Uplink Diagrammen.

Die meisten Downlink-Nachrichten werden in Frankfurt empfangen. Dort ist das Verhältnis zwischen empfangenen und lesbaren Nachrichten am schlechtesten. Die Daten aller Empfänger zusammen ergeben die Anzahl der gleichzeitig beobachteten Flugzeuge (letztes Diagramm). Das gesamte ACAS-Monitor-Netz beobachtet bis zu 1000 verschiedene Flugzeuge gleichzeitig.

ACAS-Recording 2008-Sep-05 - Mode S Radio Load Statistics

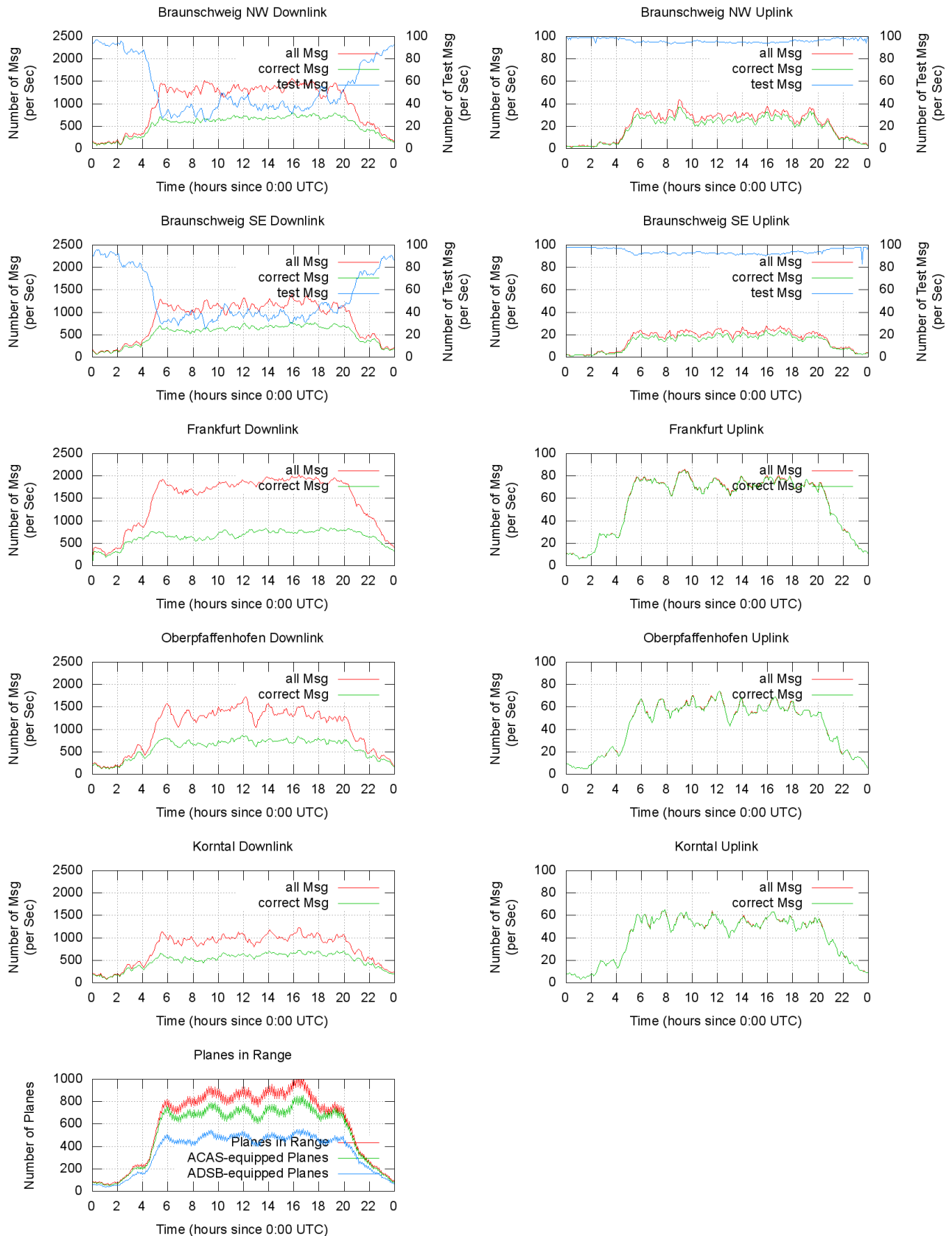


Bild 13: Funkfeldstatistik der Empfangsanlagen des ACAS-Monitor-Netzes in Braunschweig (NW und SO), Frankfurt, Oberpfaffenhofen und Korntal

4.4 Reichweite einer Station

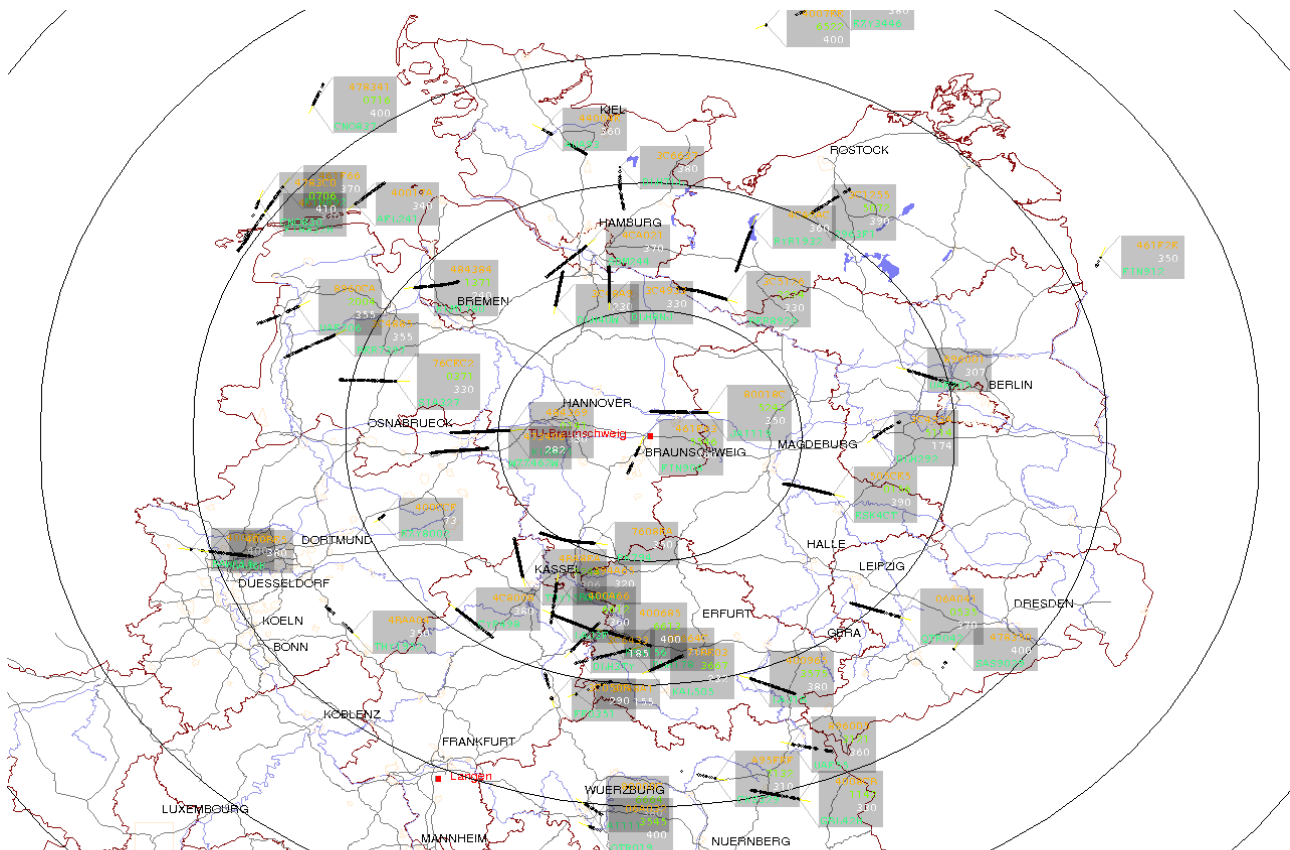


Bild 14: Reichweite der ADS-B-Station in Braunschweig

Bild 14 zeigt eine kurze Aufzeichnung von ADS-B-Spuren mit der Thales 1090MHz-Empfangsanlage und einer Omnidirektionalantenne auf dem Dach des Institutes in Braunschweig. Die Spuren der Flugzeuge sollen einen Eindruck von der Reichweite einer Station vermitteln, die Beschriftungen an den Spuren sind nicht von Bedeutung. Die Abstandskreise sind jeweils 50 NM auseinander. Die Station kann über 200 NM entfernte Ziele noch erfassen, wenn sie sich hoch genug befinden und nicht hinter dem Radiohorizont verschwinden.

5 Analyse von Mode S-Nachrichten zum Aufbau eines umfassenden Umgebungsmodells

Dieses Kapitel beschreibt den Kern der ACAS-Monitor-Software. Mit den Empfangsanlagen (Kapitel 4) werden Mode S-Nachrichten empfangen. Die ACAS-Monitoring-Idee aus Kapitel 2 muss nun in einer Software umgesetzt werden, die die empfangenen Mode S-Nachrichten analysiert und Berichte zu ACAS-Eingriffen extrahiert. Ein solcher Bericht über einen beobachteten ACAS-Eingriff wird als ACAS-Event-Report bezeichnet.

Als Implementierungssprache wurde C++ gewählt, weil:

- mit C++, als Compilersprache, große Datenmengen schnell verarbeitet werden können,
- C++ objektorientierte²⁴ und generische²⁵ Programmierung unterstützt,
- im Institut bereits eine Reihe von C-Bibliotheken vorhanden waren,
- für C++ viele gut dokumentierte Open Source Bibliotheken verfügbar sind,
- C++ unter Linux sehr gut unterstützt wird.

Zuerst muss die Architektur der ACAS-Monitor-Software gewählt werden.

5.1 Speichersparende Architektur – Datenstromverarbeitung

Schon vor der Entwicklung des ACAS-Monitors wurden testweise Empfangsdaten aufgezeichnet und statistisch wie in Abschnitt 4.3 ausgewertet. Es zeigte sich, dass ca. 1000 bis 2000 Nachrichten pro Sekunde je Antenne und Kanal zu erwarten sind. Bei ca. 100 Byte je Nachricht im Textformat ergibt das bis zu 690 MByte/Stunde oder 16 GByte/Tag, bei einem Netz von Stationen vervielfacht sich die Datenmenge entsprechend. Die einfachste Softwarearchitektur „Eingabe-Verarbeitung-Ausgabe“ lässt sich hier also nicht umsetzen, da nicht alle Daten von relevanten Zeiträumen auf einmal in den Hauptspeicher eingelesen und gehalten werden können.

Die ACAS-Monitor-Software muss aber auch gar nicht alle Nachrichten auf einmal im Speicher halten, da die Mode S-Daten von einer Empfangsanlage als zeitlich geordneter Datenstrom von Nachrichten geliefert werden und auch so verarbeitet werden können: Die Mode S-Nachrichten

²⁴ Die objektorientierte Programmierung (kurz OOP) [OOP] ist ein Programmierparadigma. Dieses Paradigma kann in dieser Arbeit nicht erklärt werden. Hier daher nur eine „Erinnerungshilfe“: In objektorientierten Datentypen (sog. Klassen) wird nicht nur die Speicherung der Daten beschrieben, sondern auch alle Funktionen (sog. Methoden), die diese Daten (Attribute) manipulieren können, d. h. Datenstruktur und dazugehörige Methoden werden zu Klassen zusammengefasst. Die Klassen sind die Datentypen („Vorlagen“) für Objekte. Ein Objekt ist eine objektorientierte Variable, die Daten speichert und Verweise auf Funktionen, die diese Daten manipulieren können, hält. Ein Beispiel: Die Klasse Konto legt fest, dass zu jedem Konto eine Kontonummer, ein Inhaber und ein Guthaben gehört, außerdem legt die Klasse Konto fest, dass ein Konto nur mit den Methoden Gutschreiben und Abbuchen verändert werden kann. Zwei Beispiele für Objekte (Variablen) der Klasse Konto sind das Konto von Alice (Kontonummer: 42) und das Konto von Bob (Kontonummer: 56). Da diese Objekte zur Klasse Konto gehören, können nur die Methoden Gutschreiben und Abbuchen die Daten verändern. Dies erleichtert die Programmierung ungemein, weil der Programmierer leichter überblickt, wo welche Daten manipuliert werden.

²⁵ Generische Programmierung ist ein Verfahren zur Entwicklung wiederverwendbarer Software. Dabei werden Funktionen möglichst allgemein entworfen, um für unterschiedliche Datentypen und Datenstrukturen verwendet werden zu können. Generische Algorithmen werden nicht für einen bestimmten Datentyp geschrieben, sondern stellen nur bestimmte Anforderungen an die Typen. Die Implementierung erfolgt bei C++ durch das Konzept der Templates. [C++]

werden in der Reihenfolge, wie diese empfangen wurden, an die ACAS-Monitor-Software weitergegeben. Die Software betrachtet jede einzelne Nachricht, extrahiert und speichert nur die relevanten Informationen aus der Nachricht in einem Umgebungsmodell (Wissensbasis). Die Software entfernt die gerade gelesene Nachricht aus dem Speicher und wendet sich der nächsten zu. Wird ein Hinweis zu einem ACAS-Eingriff gefunden, so wird aus dem Umgebungsmodell ein Bericht erstellt und ausgegeben. Das Umgebungsmodell wird von Zeit zu Zeit von veralteten Informationen bereinigt, um Speicherplatz zu sparen.

5.2 Umgebungsmodell – Klasse `Plane_DB` und ihre Helfer

Kern dieses Ansatzes ist das Umgebungsmodell, das den Zustand aller erfassten Flugzeuge im Luftraum wiedergibt, es ist also eine „Flugzeug-Status-Datenbank“, die im C++-Quelltext des ACAS-Monitors und im Folgenden als `Plane_DB` bezeichnet wird. Es ist nun die Datenstruktur des Umgebungsmodells zu entwickeln und zu klären, welche Daten einfließen sollen.

In Kapitel 2 wurde festgestellt, dass folgende Nachrichten mitgehört werden müssen:

- (1) DF11-Squitter (und die Verwandten DF17-Extended-Squitter, UF16-32-ACAS-Broadcast) enthalten `000000hex` (`FFFFFFhex` bei UF16-32) im AP-Feld bzw. PI-Feld und können somit ohne weiteres Wissen geprüft werden. Die Squitter liefern die Mode S-Adressen aller erfassten Flugzeuge in der Umgebung. Diese bilden die Grundlage eines Umgebungsmodells.
- (2) DF00-Air-Air-Surveillance-Replies, DF04-Altitude-Replies und DF17 enthalten die Höhen und Positionsdaten jedes Flugzeuges und müssen somit Teil des Umgebungsmodells werden.
- (3) DF05-Identify-Replies liefern die Zuordnung der Squawks zu Mode S-Adressen, die Squawks sind Attribute von Flugzeugen und müssen in das Umgebungsmodell einfließen.
- (4) UF16/DF16-Long-Air-Air-Surveillance erzeugen einen detaillierten Mitschnitt der eigentlichen ACAS-Kommunikation, um das RA (Resolution Advisory) und den Ablauf der Koordination zu rekonstruieren. Auch diese ACAS-Zustände könnten Teil des Umgebungsmodells sein. Da aber die ACAS-Dialoge extrahiert werden sollen, wird die ACAS-Kommunikation gleich in einer gesonderten Datenstruktur, der `ACAS_Events_DB` gespeichert. So wird das Ausgeben der Information erleichtert (siehe dazu auch Kapitel 6).

5.2.1 Von der Adressliste zum Umgebungsmodell `Plane_DB`

In Abschnitt 2.4 wurde ein Algorithmus zur Prüfung von Nachrichten auf Übertragungsfehler entwickelt. Zur Erinnerung: Das AP-Feld einer Nachricht enthält gleichzeitig die 24-Bit-Mode S-Adresse der Nachricht und auch die 24-Bit-CRC-Prüfsumme der Nachricht. Nach Auflösen der CRC-Prüfsumme bleibt im AP-Feld der Absender (bei Downlink-Nachrichten) bzw. Adressat (bei Uplink-Nachrichten). Eine Nachricht kann also dann als korrekt übertragen angesehen werden, wenn das AP-Feld die Adresse eines Flugzeuges enthält, das sich gerade im Empfangsbereich befindet. Dazu muss die Software erfahren, welche Flugzeuge im Empfangsbereich sind.

DF11- und DF17-Squitter enthalten im PI-Feld `000000hex`, das bei diesen Formaten auch als PI-Feld bezeichnet wird, und lassen sich somit ohne weiteres Wissen prüfen. Im Nachrichtenkörper

enthalten sie die Mode S-Adresse des Absenders. Aus den Squittern muss eine Liste von Mode S-Adressen von Flugzeugen in der Umgebung erzeugt werden. Gegen diese Liste müssen dann die AP-Felder aller eingehenden Nachrichten geprüft werden.

Damit die Liste nicht mit der Zeit alle jemals gesichteten Mode S-Adressen aggregiert und gestörte Nachrichten als richtig angesehen werden, muss die Liste von Flugzeugen, die nicht mehr im Empfangsbereich sind, gesäubert werden. Dazu muss zu jeder Adresse der Zeitpunkt des letzten empfangenen Squitters gespeichert werden. Hat sich ein Flugzeug länger nicht gemeldet, ist es wohl nicht mehr im Empfangsbereich und wird aus der Liste entfernt. In der jetzigen Implementierung ist diese Zeit auf 10 min gesetzt, so dass kurzzeitige Abschattungen nicht zum Löschen des Flugzeuges führen.

Neben dem Zeitpunkt des letzten Squitters können auch alle anderen Daten, die einem Flugzeug zuzuordnen sind, mit in der Adressliste gespeichert werden. Aus der Adressliste wird somit das komplette Umgebungsmodell (im Quelltext als Klasse Plane_DB). Es bildet die zentrale Wissensbasis der ACAS-Monitor-Software, die alles Wissen zu jedem Flugzeug speichert. Plane_DB ist damit als eindeutige Zuordnung von Mode S-Adressen auf Flugzeugdatensätze (Klasse Plane) aufzubauen.

Da bei der CRC-Prüfung das AP-Feld **jeder** Mode S-Nachricht in der Plane_DB gesucht werden muss, sollte diese Operation möglichst effizient sein. In der Plane_DB wird daher eine STL²⁶-Map²⁷ benutzt, um die Zuordnung von Mode S-Adressen zu Plane-Objekten zu realisieren, d. h. Mode S-Adressen bilden die Indizes der Map, Plane-Objekte die Datensätze. Die Suche in einer STL-Map hat eine Laufzeitkomplexität von $O(\log n)$ ²⁸ bei n Flugzeugen im Luftraum. Die C++-Definition der Plane_DB ist wie folgt (vereinfacht):

```
class Plane_DB {
    std::map< ModeSAdresse, Plane > db;
    ...
}
```

Bild 15 zeigt das UML-Klassendiagramm [UML] der Plane_DB zusammen mit den dazugehörigen Helfer-Klassen Plane, Track und OnePos.

5.2.2 Methoden der Klasse Plane_DB

Plane_DB selbst ist für das Speichern und den Zugriff auf Plane-Objekte und die CRC-Prüfung einer jeden eingehenden Nachricht verantwortlich. Die dazu nötigen Methoden sind im folgenden Abschnitt aufgeschlüsselt.

26 STL (Standard Template Library) ist die C++ Standard Bibliothek [STL]

27 Map ist eine Container Klasse der STL, die ein sortiertes assoziatives Array (Wörterbuch) bereitstellt.

28 $O(\log n)$: Landau Symbole [Knuth1] werden in der Informatik verwendet, um das asymptotische Verhalten von Funktionen und Folgen zu beschreiben. Insbesondere werden mit Landau Symbolen die Laufzeiten von Algorithmen in Abhängigkeit von der zu

verarbeitenden Datenmenge n beschrieben. Die Laufzeit $t(n)$ ist in $O(g(n))$ wenn: $0 \leq \limsup_{n \rightarrow \infty} \left| \frac{t(n)}{g(n)} \right| \leq \infty$. Im obigen Fall

bedeutet $O(\log n)$ vereinfacht: Die Suche dauert konstante Zeit länger, wenn sich die Anzahl der Schlüssel verdoppelt. Damit wird die Map auch nicht viel langsamer, wenn die Anzahl der Flugzeuge drastisch steigt.

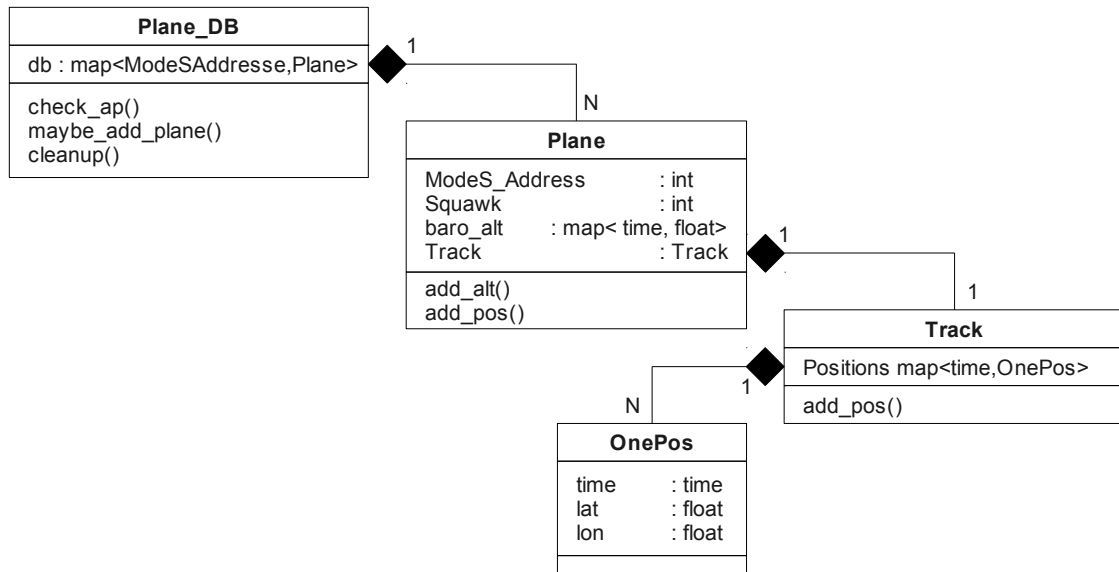


Bild 15: Plane_DB Klassenhierarchie

CRC-Prüfung für jede Nachricht

Check_ap()²⁹ implementiert die Validation (CRC-Prüfung auf Übertragungsfehler, Abschnitt 2.4) aller eingehender Mode S-Nachrichten. Die Nachricht wird an check_ap(mode_s_msg m) als Parameter m übergeben. Check_ap() prüft, ob das AP-Feld eine Mode S-Adresse enthält, die zuvor durch einen korrekten Squitter bekannt gemacht wurde. Ist diese Prüfung erfolgreich, so kann mit hoher Sicherheit davon ausgegangen werden, dass jedes Bit der Nachricht richtig übertragen wurde. Nachrichten, die einer der folgenden Eigenschaften (Tabelle 2) entsprechen, werden akzeptiert:

Eigenschaft	Bemerkung
Nachricht ist DF17 und AP-Feld = 000000 _{hex}	DF17 extended Squitter
Nachricht ist DF11 und AP-Feld = 000000 _{hex}	DF11 Squitter
Nachricht ist DF11 und Bits 1 bis 20 des AP-Feld = 00000 _{hex}	DF11 Mode S-All-Call Reply Diese DF11 wird als Antwort auf Mode S-Radar DF11-All-Calls verwendet und enthält in den Bits 21 bis 24 des AP-Feldes die Identifikation des abfragenden Radars. Laut [ICAO Annex10v4] können auch 7-Bit-Radar-IDs verwendet werden, in Deutschland ist dies aber nicht der Fall. 7-Bit-IDs können vom ACAS-Monitor unterstützt werden, sind aber zur Zeit abgeschaltet.
Nachricht ist irgendein Uplink Format mit AP-Feld = FFFFFFFF _{hex}	Im Uplink gibt es sog. Broadcasts die im AP-Feld FFFFFFFF _{hex} haben.
Plane_DB enthält ein Plane-Objekt mit derselben Adresse wie im AP-Feld	AP-Feld der Nachricht enthält die Adresse eines Flugzeuges, das bereits in der Plane_DB eingetragen ist.

Tabelle 2: Nachrichten mit diesen Eigenschaften werden als „richtig Übertragen“ angesehen

²⁹ Schreibweise check_ap(): Die Klammern kennzeichnen eine Funktion, diese Schreibweise wird in Dokumentationen (vor allem im Umfeld von C und Java) häufig verwendet.

Nur Nachrichten mit korrektem AP-Feld werden weiter verarbeitet. Alle Mode S-Nachrichten, bei denen die Prüfung fehlschlägt, werden verworfen.

Verarbeitung von Squittern – *maybe_add_plane()*

Maybe_add_plane() verarbeitet die Squitter und baut die *Plane_DB* auf, dazu extrahiert es aus DF11 oder DF17 mit AP-Feld 000000_{hex} die Mode S-Adresse des Absenders. Ist der Absender noch nicht in der *Plane_DB* vorhanden, so wird er eingetragen und ein Plane-Objekt angelegt. Unabhängig davon wird die Variable *last_seen* des Plane-Objektes auf die Empfangszeit der Mode S-Nachricht gesetzt. Damit wird das Flugzeug als noch im Empfangsbereich befindlich gekennzeichnet. Zusätzlich werden auch UF16-32-ACAS-Broadcasts mit FFFFFFF_{hex} im AP-Feld genutzt, um Adressen zur *Plane_DB* hinzuzufügen und um *last_seen* zu aktualisieren, damit steht auch ein „Squitter“ im Uplink zur Verfügung.

Entfernen alter Plane-Objekte – *cleanup_now()*

Während *maybe_add_plane()* neue Flugzeuge zur *Plane_DB* hinzufügt, entfernt *cleanup_now()* Flugzeuge, die sich eine Weile nicht mit einem Squitter gemeldet haben. Beim Aufräumen muss die komplette *Plane_DB* durchlaufen werden, was aufwändig ist. Daher wird die *Plane_DB* nicht ständig, sondern nur in regelmäßigen Abständen aufgeräumt (10 min haben sich als praktikabel erwiesen). *Cleanup_now()* prüft für jedes Plane-Objekt *p*, ob *p.last_seen* + 10 min kleiner als die aktuelle Zeit ist. Ist das der Fall, so hat sich das Flugzeug längere Zeit nicht mit einem Squitter gemeldet und scheint somit nicht mehr im Empfangsbereich zu sein. Es wird aus *Plane_DB* gelöscht.

5.2.3 Der Datensatz eines Flugzeuges – Klasse Plane

Die Objekte der Klasse *Plane* (Bild 15) bilden die Datenfelder in der *Plane_DB*. Sie speichern den aktuellen Zustand jedes im Luftraum befindlichen Flugzeuges. Bei der Verarbeitung der Mode S-Nachrichten fallen ständig Informationen über Flugzeuge an, diese Informationen werden in *Plane*-Objekten für die weitere Verarbeitung gespeichert.

Plane speichert unter anderem die Mode S-Adresse (in *ModeSAdresse*), den Squawk (in *Squawk*) und die barometrische Höhe als Höhenprofil (in *Baro_alt*). Es gibt weitere Attribute in *Plane*, die für spezielle Anwendungen hinzugefügt wurden und werden, z. B. ob das Flugzeug mit ACAS ausgerüstet ist, ob es schon einmal nicht-standard-konforme Nachrichten gesendet hat, usw..

Die in *Plane* gespeicherten Informationen lassen sich verknüpfen, um semantische Fehler in Mode S-Aussendungen zu identifizieren. Bei semantischen Fehlern ist zwar die einzelne Mode S-Nachricht korrekt formuliert empfangen worden, die Nachrichtenfolge ergibt aber keinen Sinn. Zum Beispiel werden einzelne falsche ADS-B-Positionsmeldungen von einer Spurbildung erkannt und verworfen (Abschnitt 7.5).

5.3 Nachrichtenweise Verarbeitung – Hauptschleife

Das im letzten Abschnitt entwickelte Umgebungsmodell, die *Plane_DB*, muss mit Mode S-Nachrichten versorgt werden. Dazu muss um die *Plane_DB* herum die Hauptschleife der ACAS-Monitor-

Software (Bild 16) entwickelt werden. Hier wird nun der Weg der einzelnen Mode S-Nachricht durch die Verarbeitung entwickelt und die für die Verarbeitung noch nötigen Datenstrukturen (Klassen) erläutert.

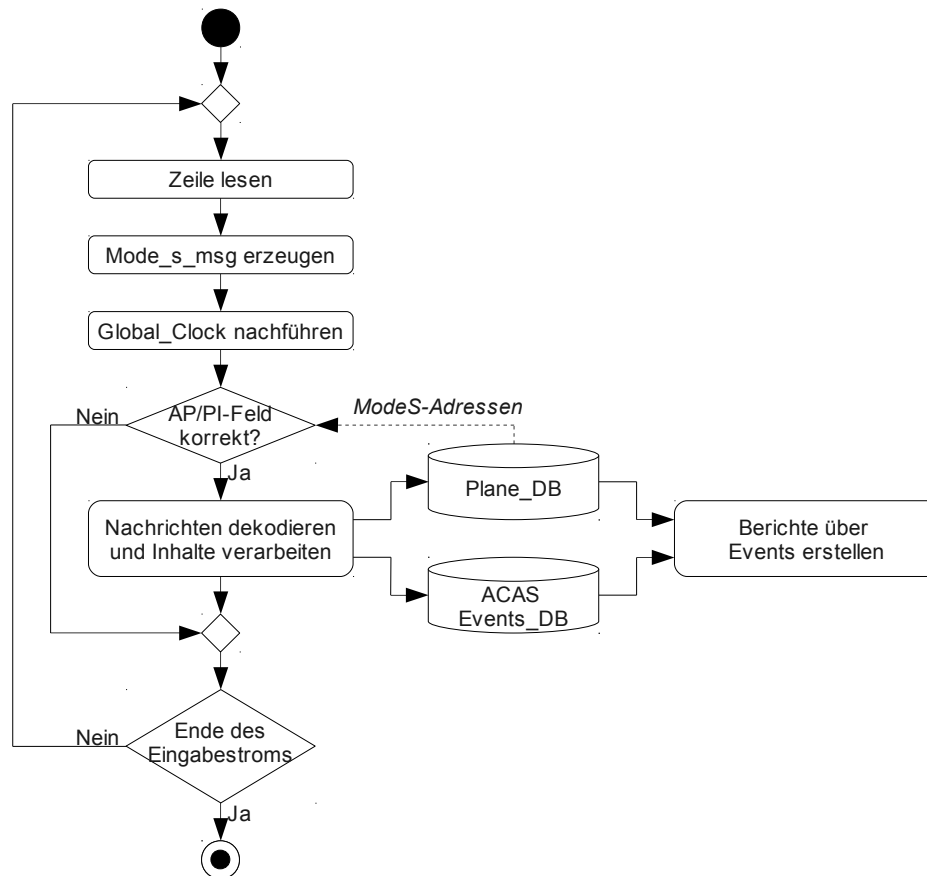


Bild 16: Detaillierte Hauptschleife

Die Mode S-Nachrichten werden der ACAS-Monitor-Software im Textformat (Thales oder TU Braunschweig, siehe Abschnitt 4.1.1) als Eingabestrom per Standardeingabe zugeführt. In den Textformaten enthält jede Zeile genau eine Nachricht, jeder Durchlauf durch die Hauptschleife bearbeitet eine Nachricht vom Eingabestrom.

Da es nicht effizient ist, im gesamten Programm Mode S-Nachrichten als Textzeilen (Strings) darzustellen und bei jeder Verwendung zu parsen³⁰, wird jede Textzeile sofort in ein Objekt der Klasse `Mode_s_msg` überführt. Ein `Mode_s_msg`-Objekt speichert eine Nachricht und ermöglicht den einfachen und effizienten Zugriff auf das AP-Feld, auf das Format, und auf einzelne oder Gruppen von Datenbits, usw. (siehe auch Abschnitt 5.4).

Mode S-Nachrichten werden von den Empfangsanlagen mit einem Zeitstempel versehen. Dieser wird von der ACAS-Monitor-Software benutzt, um die interne Uhr `Global_Clock` (Abschnitt 5.5) nachzuführen. So verfügt die ACAS-Monitor-Software immer über eine Zeit, die zu den Daten passt, egal ob die Daten online verarbeitet werden oder eine Aufzeichnung nachträglich analysiert wird.

³⁰ Parsen ist in der Informatik die Zerlegung und Umwandlung einer beliebigen Eingabe in ein für die Weiterverarbeitung brauchbares Format.

Im nächsten Schritt wird die bereits von den Empfangsanlagen berechnete 24-Bit-Prüfsumme im AP- bzw. PI-Feld nach dem Ansatz von Abschnitt 5.2.2 geprüft. D. h. Squitter mit dem PI-Feld 000000_{hex} und besondere Uplink-Formate mit FFFFFFF_{hex} im AP-Feld werden akzeptiert. Bei allen anderen Formaten wird in der Plane_DB nachgeschlagen, ob das AP-Feld zu einem Flugzeug im Empfangsbereich passt. Nachrichten, welche die CRC-Prüfung nicht bestehen, werden verworfen.

Danach folgt die eigentliche Decodierung und Interpretation der Mode S-Nachrichten. Bei Squittern wird die Mode S-Adresse extrahiert und in die Plane_DB aufgenommen, bzw. der Zeitpunkt des letzten empfangenen Squitters zu dieser Adresse aktualisiert. Für alle anderen Formate existieren interpret_X()-Funktionen (Abschnitt 5.6), die Informationen aus den Mode S-Nachrichten in die Plane_DB und in die Liste aller ACAS-Events (ACAS_Events_DB, Kapitel 6) übertragen. Dabei werden zum Beispiel Positionsdaten aus DF17 in die Plane_DB geschrieben und zu Spuren (Tracks) zusammengesetzt (Kapitel 7).

5.4 Klasse Mode_s_msg – Mode S-Nachricht

In der Klasse Mode_s_msg werden Mode S-Nachrichten gespeichert und verarbeitet. Direkt nach dem Lesen einer Nachricht von der Standardeingabe wird der Eingabe-String in eine Mode_s_msg umgewandelt. Damit muss die Nachricht nur einmal gelesen werden und alle weiteren Schritte der Interpretation können einfach auf Mode_s_msg-Objekte zugreifen.

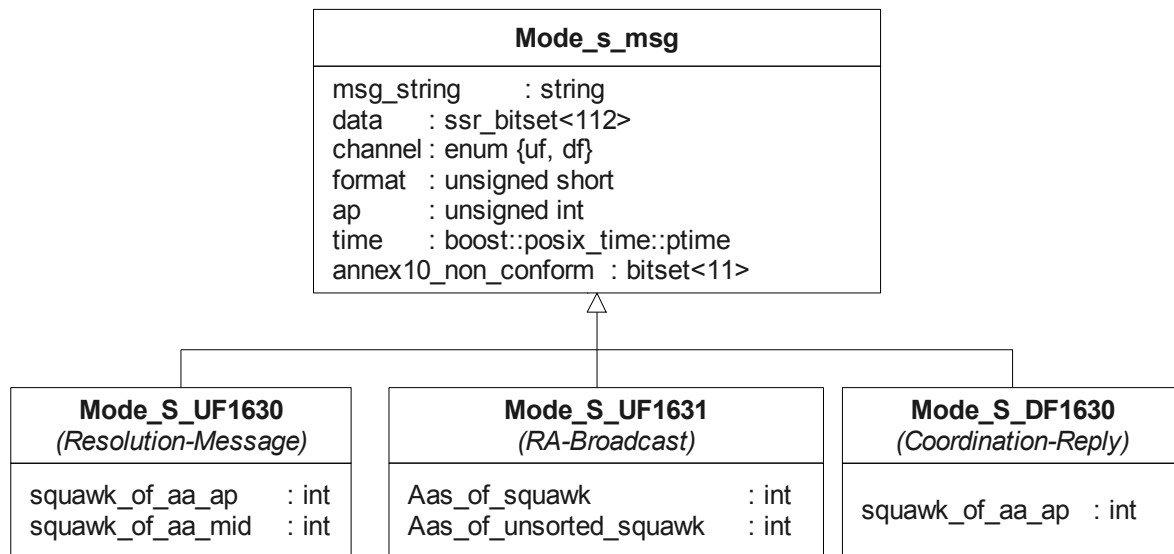


Bild 17: Klasse Mode_s_msg und abgeleitete Klassen

Bild 17 zeigt das Klassendiagramm von Mode_s_msg und die davon abgeleiteten Klassen³¹. Für die meisten Formate ist die allgemeine Klasse Mode_s_msg ausreichend, lediglich die Formate UF16-30, UF16-31 und DF16-30 müssen abgeleitet werden (siehe Kapitel 6). Tabelle 3 erläutert die Attribute der Klasse Mode_s_msg.

³¹ Das Ableiten von Klassen ist eine Standardtechnik der objektorientierten Programmierung. Eine abgeleitete Klasse erbt alle Attribute und Methoden der Basisklasse und ergänzt diese um weitere Attribute oder Funktionen. Funktionen, die für eine Basisklasse implementiert wurden, gehen mit Objekten der abgeleiteten Klasse um, als seien sie von der Basisklasse. So wird der Quelltext, der für alle Klassen gleich ist, in der Basisklasse gehalten. Spezialisierungen werden in den abgeleiteten Klassen implementiert.

Ssr_bitset

Nach dem Standard des Sekundärradars werden Bitfelder von 1 bis n indiziert. In C++, wie in den meisten Programmiersprachen, von 0 bis $n-1$. Um den Code wartbar und verständlich zu halten, wurde um das C++-STL-Bitset< n > der Wrapper Ssr_bitset< n > geschrieben, der die Indizes von 1 bis n auf 0 bis $n-1$ abbildet.

Attribut	Bedeutung
msg_string	Zeichenkette, aus der die Mode_s_msg konstruiert wurde
data	ssr_bitset<112> zum Zugriff auf einzelne Bits der Nachricht; Nummerierung der Bits wie in [ICAO Annex10v4]; 56 Bit lange Nachrichten belegen nur Bits 1 bis 56, die anderen Bits sind 0
channel	Kanal der Übertragung, df ~ Downlink, uf ~ Uplink
format	Formatnummer der Nachricht
ap	AP-Feld nach Auflösung der CRC-Prüfsumme, bei den meisten Formaten ist das die Adresse des Absenders bzw. Adressaten
time	Empfangszeitpunkt der Nachricht; der Tag richtet sich nach der Global_Clock, die Sekunden seit Mitternacht werden aus dem Zeitstempel der Nachricht übernommen.
annex10_non_conform	Fehlerbits; wird bei der Verarbeitung in einer Mode S-Nachricht ein Fehler entdeckt, so wird das entsprechende Fehlerbit gesetzt (siehe 5.4.1).

Tabelle 3: Attribute der Klasse Mode_s_msg

5.4.1 Fehlerbits – annex10_non_conform

In vielen Nachrichten werden bei der Interpretation Fehler entdeckt. Um die Fehler zu dokumentieren und um mit ihnen in der weiteren Verarbeitung umgehen zu können, müssen fehlerhafte Nachrichten durch Setzen entsprechender Fehlerbits markiert werden. Die Fehlerbits sind als eine Bitfolge mit Index (in C++ STL-Bitset) implementiert, in der auch mehrere Fehlerbits gleichzeitig gesetzt werden können. Tabelle 4 zeigt die Fehlerbits und deren Bedeutung.

Bei schwerwiegenden Fehlern kann aus der Nachricht keine weitere Information gewonnen werden, z. B. wenn die CRC-Prüfung Übertragungsfehler anzeigt. Bei weniger bedeutenden Fehlern kann die Nachricht immer noch ausgewertet werden. Bei einigen Nachrichtentypen (z. B. UF16-31) ist man sogar auf die Auswertung von fehlerhaften Nachrichten angewiesen, um auch Nachrichten von nicht vollständig Standard-konformen ACAS-Geräten auszuwerten.

Meist führt das Erkennen von Fehlern lediglich zum Setzen von Fehlerbits und nicht dazu, dass Nachrichten sofort verworfen werden. Fehlerhafte Nachrichten werden an die weiteren Verarbeitungsstufen weitergereicht, wo dann im Zusammenhang anhand der Fehlerbits entschieden werden kann, ob die Nachricht ausgewertet wird oder nicht. Lediglich bei Übertragungsfehlern, wenn die CRC-Prüfung einer Nachricht fehlschlägt (Fehlerbit 0), wird die Nachricht sofort verworfen.

Wird der ACAS-Monitor weiterentwickelt, so können (und müssen) auch weitere Fehlerbits definiert werden.

Index der Fehlerbits	Bedeutung
0	CRC-Check fehlgeschlagen, d. h. AP-Feld enthält keine bekannte Mode S-Adresse und auch keine andere gültige Prüfsumme.
1	Nachricht enthält ungültigen Squawk, d. h. Bit 7 (sog. X-Bit) ist gesetzt, obwohl [ICAO Annex10v4] in diesem Nachrichtenformat eindeutig eine 0 als X-Bit verlangt.
2	Das AP-Feld der UF16-31 enthält nicht FFFFFFF _{hex} .
3	Fehlerhafte Höhe, nicht definierter Höhencode wurde verwendet.
4	Bisher nicht standardisierte metrische Höhe wurde gemeldet.
5	ACAS Nachrichten enthalten bisher nicht standardisierte ACAS III Informationen.
6	AP-Feld enthält 000000 _{hex} oder FFFFFFF _{hex} obwohl dieses Format kein Squitter oder Broadcast ist.
7	Der Empfänger, der in einer UF16-30 angegeben wurde, ist bisher nicht in Plane_DB enthalten. Da der Empfänger im AP-Feld genannt wird, kann dieser Fehler auch durch einen Übertragungsfehler entstehen (wobei auch Fehlerbit 0 gesetzt würde).
8	Der Absender, der in einer UF16-30 angegeben wurde, ist nicht in der Plane_DB enthalten, d. h. vom Absender wurde noch kein Squitter empfangen.
9	UF16-30 Redundanzprüfung schlägt fehl.
10	Die enthaltenen Informationen sind nicht standard-konform. Meist sind in diesem Fall noch weitere Bits gesetzt.

Tabelle 4: Bedeutung der definierten Fehlerbits

5.4.2 Von Mode_s_msg abgeleitete Klassen

Die meisten Mode S-Nachrichten werden sofort beim Empfang verarbeitet und die extrahierten Informationen in die Plane_DB geschrieben, für solche Nachrichten reicht die einfache Klasse Mode_s_msg aus. Zur Beschreibung der ACAS-Kommunikation müssen RA-relevante Mode S-Nachrichten (UF16-30, UF16-31 und DF16-30) zwischengespeichert, analysiert und dokumentiert werden. Zu den Nachrichten müssen Hintergrund-Informationen aus der Plane_DB zugeführt werden. Dazu mussten für UF16-30-Resolution-Message, UF16-31-RA-Broadcast und DF16-30-Coordination-Reply spezialisierte abgeleitete Klassen von Mode_s_msg abgeleitet werden, die die spezifischen Informationen dieser Nachrichten speichern können (Bild 17 Seite 45).

Zum Beispiel ist in einer UF16-31-RA-Broadcast der Squawk des Absenders genannt, nicht aber seine Mode S-Adresse. Beim Decodieren werden die Mode S-Adressen aller Flugzeuge mit diesem Squawk aus der Plane_DB in die Nachricht kopiert und mit der Nachricht gespeichert. So bleiben die aktuell gefundenen Mode S-Adressen gespeichert, auch wenn der Squawk neu vergeben wird oder gefundene Flugzeuge andere Squawks zugewiesen bekommen.

5.5 Analysezeit – Klasse Global_Clock

Meist werden an Empfangsanlagen Aufzeichnungen gemacht, die nachträglich von der ACAS-Monitor-Software analysiert werden. Selbst wenn die ACAS-Monitor-Software direkt an die Empfangsanlagen gekoppelt ist, so erzeugt die Weiterleitung der Nachrichten über Netzwerke und

Softwareschichten bis zur ACAS-Monitor-Software eine Totzeit von Millisekunden bis zu Sekunden. Es ist also angebracht, dass die Analysezeit unabhängig von der Systemzeit des Computers ist. Es ist naheliegend, die Analysezeit den Empfangszeitpunkten der Nachrichten folgen zu lassen. Dazu musste im Programm eine globale Uhr (Klasse `Global_Clock`) eingebaut werden, der jede eingehende Nachricht vorgelegt wird.

Boost Zeit-Bibliothek

Im gesamten Projekt werden Zeitpunkte als „`boost::posix_time::ptime`“ Objekte gespeichert [Boost]. Denn die Boost-Day-Time-Bibliothek stellt einen umfangreichen, getesteten und gut dokumentierten Satz von Zeitfunktionen zur Verfügung. Die Klasse `ptime` speichert Zeiten in 64 Bit und einer Auflösung von einer Mikrosekunde, was mehrere tausend Jahre abdeckt. Dieser große Bereich von Zeitpunkten wird von der ACAS-Monitor-Software zwar nicht ausgeschöpft, jedoch reichen alternative 32-Bit Zeit-Bibliotheken nicht aus und 48-Bit Zeitformate bringen kaum Speicherersparnis und sind nicht schneller als 64-Bit Implementierungen.

Zeitzone

Der ACAS-Monitor beachtet generell keine Zeitzone. Alle Zeitstempel werden so verarbeitet, wie sie in den Eingaben vorliegen. Die ACAS-Monitor-Software funktioniert somit in jeder Zeitzone, solange alle Zeitpunkte aus derselben Zeitzone sind. In der Datenverarbeitung der Luftfahrt wird empfohlen, nur UTC zu verwenden und lediglich die Ausgaben an lokale Bedürfnisse anzupassen.

Zeitnachführung – Tagessekunden – Datumsübergänge

Von den am IfEV vorhandenen Empfangsanlagen und deren Vorverarbeitungssoftware werden Mode S-Nachrichten mit Zeitstempeln versehen. Diese Zeitstempel sind meist Sekunden seit 00:00 Uhr (Tagessekunden). Da ACAS-Monitoring auch über 00:00 hinweg arbeiten soll, muss zwischen 0:01 Uhr am Tag A und 0:01 Uhr am Tag A+1 unterschieden werden. Die ACAS-Monitor-Software tut dies auf einfache Weise:

Beim Start der Analyse wird das aktuelle Datum des Computers verwendet oder der ACAS-Monitor-Software wird ein Datum beim Aufruf mitgeliefert. Der Zeitstempel der ersten Mode S-Nachricht legt dann die Simulationsuhrzeit (`Global_Clock`-Zeit) fest. Jede eingehende Mode S-Nachricht muss von `Global_Clock` betrachtet werden, dabei werden drei Fälle unterschieden:

1. Normalfall:

Jede eingehende Nachricht wird mit dem `Global_Clock`-Datum versehen, die Simulationsuhr `Global_Clock` wird auf den Zeitstempel der Nachricht gesetzt, wenn dieser größer als die `Global_Clock`-Zeit ist.

2. Ausnahme A: `Global_Clock > 23:00` und `Zeitstempel < 01:00`

Ist die `Global_Clock`-Zeit größer als 23.00 Uhr und der Zeitstempel der eingehenden Nachricht hingegen kleiner als 01:00 Uhr, so muss ein Datumsübergang vorliegen. Das Datum wird um eins erhöht und `Global_Clock` entsprechend gesetzt.

3. Ausnahme B: Global_Clock < 01:00 und Zeitstempel > 23:00

Ist die Global_Clock-Zeit kleiner als 01:00 Uhr und der Zeitstempel größer als 23:00 muss es sich um eine verspätete Nachricht vom Vortag handeln. Das Datum der Nachricht wird auf den Vortag gesetzt, die Global_Clock wird nicht verändert.

Mögliches Übersehen von Tagesübergängen

Diese Art der Zeitnachführung funktioniert, wenn mindestens einmal pro Stunde Mode S-Nachrichten empfangen werden und Nachrichten nicht länger als eine Stunde verzögert werden. Wenn die Empfangsanlage ausfällt oder keine Flugzeuge mit Mode S im Empfangsbereich sind, so kann es dazu kommen, dass längere Zeit keine Nachrichten an den ACAS-Monitor weitergereicht werden und ein Tagesübergang nicht erkannt wird. In den Empfangsanlagen in Braunschweig werden zusätzlich zu den von Flugzeugen ausgesendeten Mode S-Nachrichten auch eigene Testsignale in die Empfangsanlagen eingespeist (siehe Abschnitt 4.3.1). In den anderen Thales-Empfangstationen wird per Software jede Sekunde eine Testnachricht in den Datenstrom eingefügt. Dies stellt sicher, dass ständig Nachrichten an den ACAS-Monitor geliefert werden.

5.6 Funktionen interpret_X() und Verwandte

Die eigentliche Decodierung der Mode S-Nachrichten erfolgt durch die Funktionen interpret_X(), wobei X je nach Aufgabe der Funktion variiert. Jeder interpret_X() Funktion wird die Mode_s_msg oder der Teil einer Mode S-Nachricht übergeben, der interpretiert werden soll. Die wichtigste Funktion ist interpret_msg(). Sie verzweigt je nach Format der übergebenen Nachricht. Die dann aufgerufenen Funktionen sind interpret_DF00(), interpret_UF16(), interpret_DF17(), interpret_DF05() und andere. Mit ihnen werden die anwendungsspezifischen Aufgaben wie ADS-B-Decodierung (Abschnitt 7.4) oder ACAS-Monitoring (Kapitel 6) implementiert. Zum Beispiel extrahiert interpret_DF00() die Höhe aus einer DF00 und trägt sie in die Plane_DB ein. Interpret_DF16() ruft zuerst interpret_DF00() auf, damit der DF00-Teil der DF16 interpretiert wird, dann wird der ACAS-relevante Teil vom ACAS-Event-Monitoring (siehe nächstes Kapitel) interpretiert.

6 Zusammenstellen von ACAS-Nachrichten zu ACAS-Event-Berichten

Im letzten Kapitel wurde gezeigt, wie ein Datenstrom von Mode S-Nachrichten analysiert werden kann, um die einzelnen Nachrichten zu validieren und um ein Umgebungsmodell (Wissensbasis) aufzubauen. Ausgehend davon lassen sich verschiedene Analysen betreiben. In dieser Arbeit wird ein ACAS-Monitor entwickelt und damit nach Resolution Advisories (RAs) gesucht. Die Softwaremodule des letzten Kapitels wurden aber auch in anderen Untersuchungen verwendet, z. B. um Flugzeuge mit veralteter Ausrüstung zu finden oder um ADS-B-Daten zu decodieren und zu filtern.

Der ACAS-Monitor soll nicht einfach empfangene ACAS-Kommunikationen ausgeben, sondern die Vorfalldesreibungen von ACAS-Eingriffen, sog. ACAS-Event-Berichte, zusammenstellen. Dazu filtert die Software RA-relevante Nachrichten aus dem Nachrichtenstrom heraus, baut diese zu sinnfälligen Folgen von Nachrichten, bzw. sinnvollen Dialogen zusammen und reichert die Nachrichtenfolgen mit Informationen aus dem Umgebungsmodell (Plane_DB) an. Solche angereicherten Folgen von ACAS-Nachrichten werden in dieser Arbeit als ACAS-Events bezeichnet. Dabei sind die Formate UF16-30-Resolution-Message, DF16-30-Coordination-Reply und UF16-31-RA-Broadcast RA-relevant (siehe auch Abschnitt 2.2) und werden von den in diesem Kapitel beschriebenen Modulen verarbeitet.

Die Idee beim Zusammenstellen von ACAS-Events ist folgende: Alle aktuellen ACAS-Events werden in der ACAS_Events_DB gespeichert. Neu empfangene ACAS-relevante Nachrichten werden zum entsprechenden Event in der ACAS_Events_DB hinzugefügt. Wenn eine ACAS-relevante Nachricht nicht zu einem bestehenden Event in der ACAS_Events_DB zuzuordnen ist, wird ein neues Event angelegt. Stellt sich heraus, dass zwei Events in der ACAS_Events_DB zu einem Vorfall gehören, weil eine Koordination zwischen den Flugzeugen beider Events stattfindet, so werden die Events vereinigt. Sind zu einem ACAS-Event eine gewisse Zeit (derzeit 5 min) keine neuen Nachrichten hinzugekommen, wird es von der Software als beendet angesehen, ausgegeben und aus der ACAS_Events_DB entfernt.

Die ausgegebenen Event-Berichte werden dann automatisch weiteren Verarbeitungsschritten mit externer Software zugeführt, d. h. sie werden nachbearbeitet, sortiert, archiviert und für verschiedene Statistiken ausgewertet.

So sind die ACAS_Event-Objekte die Modelle der gerade im Luftraum ablaufenden ACAS-Eingriffe.

6.1 Speicherung von ACAS-Events – die Klassen ACAS_Events_DB und ACAS_Event

Die in Bild 18 gezeigte Klasse ACAS_Events_DB speichert alle gefundenen ACAS-Events. ACAS_Events_DB hält die Liste aller ACAS-Events und eine Liste mehrdeutiger UF16-31 (siehe

Abschnitt 6.2.2). Events, für die in den letzten `time_till_erased` Sekunden keine neuen Nachrichten empfangen wurden, werden ausgegeben und aus `ACAS_Events_DB` entfernt.

Ein `ACAS_Event`-Objekt speichert alles Wissen zu einem ACAS-Vorfall, es ist eine Folge von RA-relevanten Mode S-Nachrichten. Demzufolge ist die Klasse `ACAS_Event` als eine Liste von `Mode_s_msg`-Objekten implementiert. Außerdem muss in einem `ACAS_Event` noch die Menge der an diesem Event beteiligten Flugzeuge (Mode S-Adressen) gespeichert werden, um nicht immer alle Nachrichten durchsuchen zu müssen.

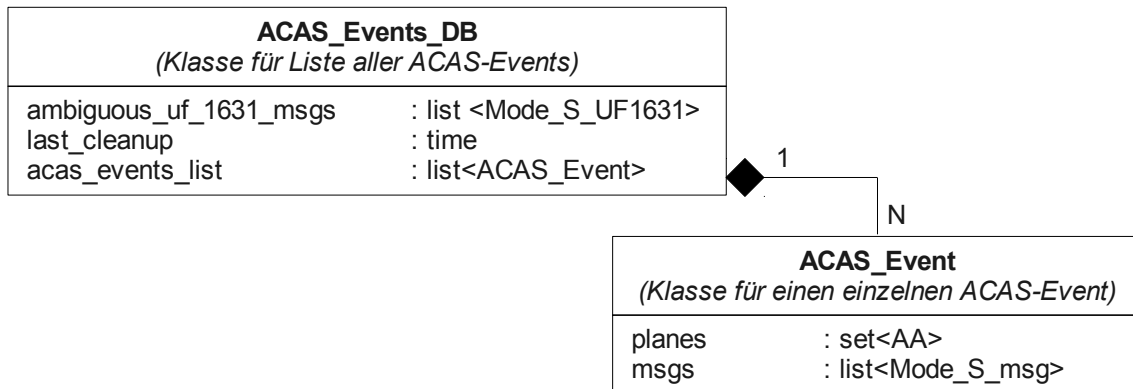


Bild 18: Klassen `ACAS_Events_DB` und `ACAS_Event`

6.2 ACAS-Nachrichten zu ACAS-Events zusammenstellen

Die RA-relevanten Nachrichten UF16-30, DF16-30 oder UF16-31 werden an die Funktion `found_acas_events()` (Bild 19) übergeben, diese ruft für jedes Nachrichtenformat eine eigene Hilfsfunktion auf, diese Hilfsfunktionen werden unten erläutert. Aufgabe der Hilfsfunktionen ist es, die RA-relevanten Nachrichten den bestehenden `ACAS_Event`-Objekten zuzuordnen oder neue `ACAS_Event`-Objekte zu erzeugen.

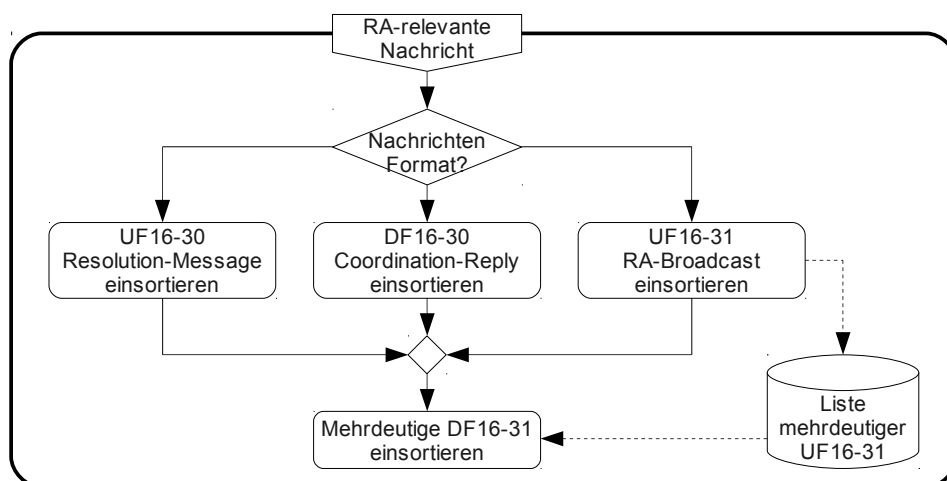


Bild 19: Überblick über die Verarbeitung RA-relevanter Nachrichten in der Funktion `found_acas_events()`

UF16-30 und DF16-30 können von den Hilfsfunktionen direkt in ACAS-Events sortiert werden. Bei UF16-31 ist das leider nicht immer sofort möglich, da UF16-31 nicht die Mode S-Adresse des Absenders, sondern einen möglicherweise mehrdeutigen Squawk enthält. Daher werden mehrdeutige UF16-31 erst einmal in einer Liste zurückgehalten. Nachdem weitere RA-relevante

Nachrichten verarbeitet wurden und somit neue Informationen in der ACAS_Events_DB vorliegen, wird versucht, bisher nicht zugeordnete mehrdeutige UF16-31 in ACAS_Events_DB einzutragen.

6.2.1 UF16-30 und DF16-30 in ACAS_Event-Objekte zuordnen

Die Verarbeitung von DF16-30 und UF16-30 verläuft ähnlich, beide Abläufe sind in Bild 20 und Bild 21 dargestellt (Routinen zum Umgang mit fehlerhaften Nachrichten wurden weggelassen).

DF16-30 zuordnen

Am einfachsten ist das Einsortieren von DF16-30-Coordination-Replies (Bild 20). Diese enthalten lediglich die Mode S-Adresse des Absenders der Nachricht. Ist bereits ein ACAS-Event in der ACAS_Events_DB mit diesem Flugzeug vorhanden, so wird die DF16-30 hinzugefügt. Ist kein Event vorhanden, an dem dieses Flugzeug beteiligt ist, so wird ein neuer ACAS-Event erstellt und in die ACAS_Events_DB eingetragen.

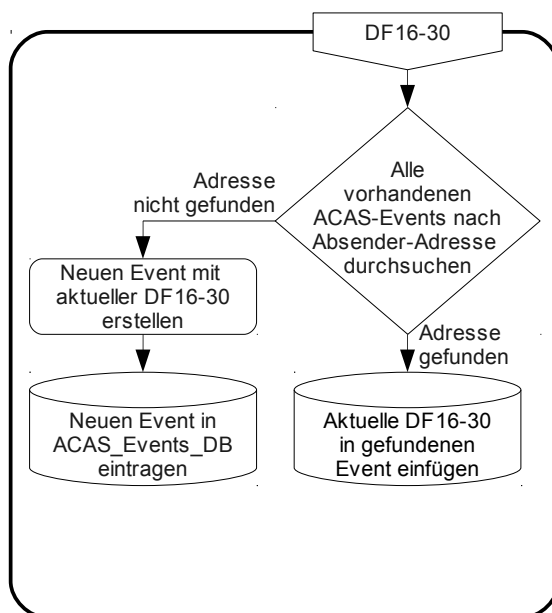


Bild 20: Verarbeitung von DF16-30

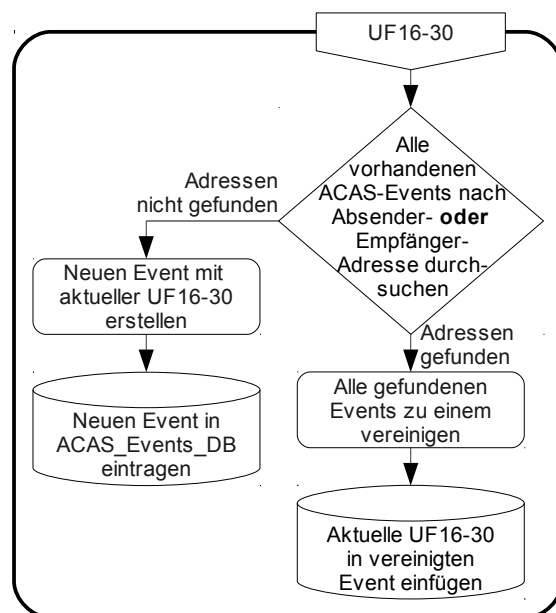


Bild 21: Verarbeitung von UF16-30

UF16-30 zuordnen

Im Gegensatz zur DF16-30 enthalten die UF16-30 Resolution Messages die Mode S-Adressen von Absender und Empfänger. Die UF16-30 ist somit die einzige Nachricht, die Informationen darüber enthält, welche Flugzeuge **gemeinsam** in einem ACAS-Event verwickelt sind. Beim Einsortieren der UF16-30 (Bild 21) werden zuerst alle Events gesucht, in denen mindestens eines der beiden Flugzeuge verwickelt ist. Dabei können die beiden Mode S-Adressen in a) keinem, b) genau einem oder c) in zwei Events gefunden werden.

- Wird kein Event gefunden, so muss eine neuer Event mit dieser Nachricht angelegt werden.
- Wird genau ein passender Event gefunden, so wird die Nachricht einfach zu diesem hinzugefügt.
- Es können auch zwei Events gefunden werden. In einen Event ist der Absender, im anderen der Empfänger der UF16-30 verwickelt. Die Events sind noch nicht als zusammengehörig

erkannt worden, weil noch keine UF16-30 empfangen wurde. Mit dem Empfang dieser UF16-30 werden die Events dann zu einem vereinigt und die UF16-30 zum Event hinzugefügt.

Fall b) und c) lassen sich wie in Bild 21 gemeinsam behandeln. Denn die Operation „Alle gefundenen Events zu einem Vereinigen“ aus Fall c) schließt auch die Vereinigung von einem Event zu einem Event wie in Fall b) ein.

Beispiel

Durch Übertragungsstörungen gehen oft einzelne Nachrichten verloren. Angenommen der ACAS-Monitor erhält folgende Nachrichten in dieser Reihenfolge:

1. DF16-30-Coordination-Reply von Flugzeug A
2. DF16-30-Coordination-Reply von Flugzeug B
3. UF16-30-Resolution-Message von Flugzeug A an Flugzeug B

Flugzeug A sendet Nachricht 1, ein DF16-30-Coordination-Reply, nur dann, wenn es mit einer UF16-30-Resolution-Message dazu aufgefordert wurde (siehe auch Bild 1). Der ACAS-Monitor hat diese UF16-30 im Beispiel aber nicht empfangen. Genauso wurde die zu Nachricht 2 gehörende UF16-30 nicht empfangen.

Beim Verarbeiten von Nachricht 1 wird ein ACAS-Event mit dem beteiligten Flugzeug A erstellt. Beim Empfang von Nachricht 2 kann nicht festgestellt werden, ob B im ACAS-Event mit A liegt oder ob es sich um einen weiteren unabhängigen ACAS-Event handelt – mit Nachricht 2 wird daher ein weiterer ACAS-Event erstellt, an dem nur B beteiligt ist. Beim Empfang von Nachricht 3 werden alle ACAS-Events, an denen A oder B beteiligt sind, zusammengefasst und Nachricht 3 an den zusammengefassten ACAS-Event angehängt.

6.2.2 UF16-31 Resolution Advisory Broadcast zuordnen

Der UF16-31-RA-Broadcast enthält keine Mode S-Adresse, sondern lediglich den Squawk des Absenders. Die Zuordnung zu ACAS-Events kann demzufolge nur indirekt erfolgen. Erschwerend kommt hinzu, dass bei der Codierung der Squawks in der UF16-31 ein Widerspruch zwischen [ICAO Annex10v4] und [RTCA DO-185A] vorliegt (Abschnitt 2.7.3), was dazu führt, dass der Squawk auf zwei Arten decodiert werden kann. Die ACAS-Monitor-Software muss notgedrungen beide unterstützen, somit liefert jede UF16-31 zwei gleichberechtigte Squawks.

Um UF16-31 zu Events zuzuordnen, müssen zuerst die Mode S-Adressen der möglichen Absender ermittelt werden. Dabei werden in der Plane_DB alle Flugzeuge gesucht, die einen der beiden Squawks per DF05 oder DF21 gemeldet haben. Diese Liste wird mit der UF16-31 gespeichert.

Unmittelbares Zuordnen

Wie bei den anderen RA-relevanten Nachrichten wird versucht, UF16-31 unmittelbar nach dem Empfang den Events zuzuordnen. Der Ablauf ist in Bild 22 dargestellt. Hat genau ein Flugzeug einen der Squawks der UF16-31, so ist die Zuordnung der Squawks eindeutig und die UF16-31 kann direkt in ein Event einsortiert werden oder ein neues Event mit dieser Adresse wird angelegt.

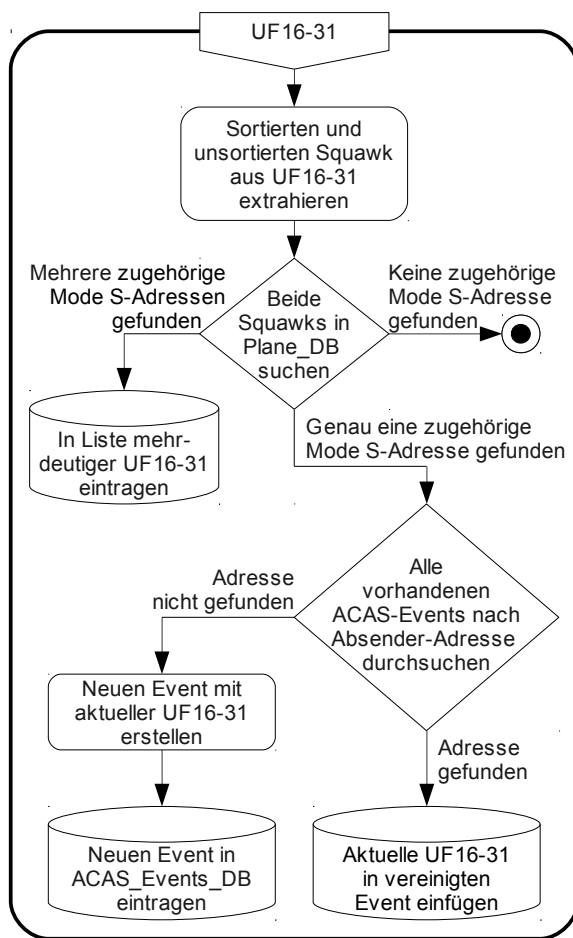


Bild 22: Unmittelbare Zuordnung von UF16-31

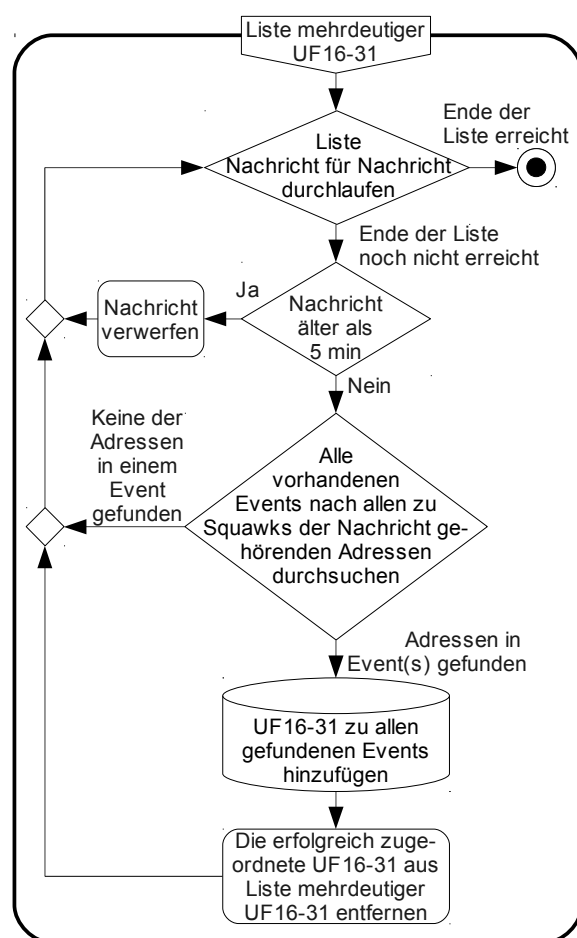


Bild 23: Nachträgliche Zuordnung mehrdeutiger UF16-32

Nachträgliches Zuordnen

Wurden mehrere Flugzeuge zu den Squawks gefunden, so hat die UF16-31 keinen eindeutigen Absender. Als letzte Möglichkeit können alle Events nach passenden Mode S-Adressen durchsucht werden. Meist wird nur einer der in Frage kommenden Absender der UF16-31 gerade in ein ACAS-Event verwickelt sein. Dieser Zuordnungs-Schritt hat keinen Erfolg, wenn die aktuelle UF16-31 die erste empfangene Nachricht zu diesem Event ist und somit noch kein passender Event in der ACAS_Events_DB angelegt ist. Daher werden alle mehrdeutigen UF16-31 zunächst in einer Liste zwischengespeichert. Nachdem eine weitere RA-relevante Nachricht verarbeitet wurde, wird jede noch nicht zugeordnete UF16-31 erneut geprüft, d. h. die Funktion von Bild 23 wird ausgeführt.

Beispiel

1. Flugzeug A sendet eine DF16-30
2. Flugzeug B sendet UF16-31 mit Squawk 7727 bzw. 3577. 3577 sei zweimal in der Plane_DB vorhanden – einmal bei Flugzug B und einmal bei Flugzeug C.
3. Flugzeug B sendet danach UF16-30 an Flugzeug D.

Beim Empfang von Nachricht 1 wird ein Event mit Flugzeug A angelegt.

Beim Empfang von Nachricht 2 kann diese noch keinem Event zugeordnet werden, denn zum einen könnten B und C Absender von Nachricht 2 sein, zum andern gibt es keinen Event, an dem einer der beiden beteiligt ist. Die Nachricht 2 wird erst einmal zurückgehalten.

Mit dem Empfang von Nachricht 3 wird ein Event angelegt, an dem B und D beteiligt sind. Danach wird die Liste der mehrdeutigen UF16-31 durchsucht, darin ist nur Nachricht 2. Nun gibt es einen Event, in dem einer der möglichen Absender von Nachricht 2 verwickelt ist. Es folgt, dass B der Absender von Nachricht 2 ist und Nachricht 2 zu dem Event zwischen B und D zugeordnet werden muss. C ist nicht beteiligt. Der zu Beginn einzige Event mit Flugzeug A hat somit gar nichts mit dem Event zwischen B und D zu tun, sondern spielt sich nur zufällig zur gleichen Zeit anderenorts im Empfangsbereich ab.

6.3 Ausgabe von Berichten

Wurden zu einem ACAS-Event einige Zeit (zur Zeit auf 5 min gesetzt) keine Nachrichten empfangen, so wird das ACAS-Event als Verzeichnis ausgegeben und aus der ACAS_Events_DB entfernt und zur Weiterverarbeitung, Sortierung und Archivierung durch weitere Programme freigegeben. Kapitel 9 erklärt den Aufbau der ACAS-Event-Verzeichnisse und zeigt Beispiele.

7 Verarbeitung von Höhen und Positionen als Zusatz zur Interpretation von ACAS-Events

Zur Interpretation von ACAS-Eingriffen sind Höhen und Positionsdaten unerlässlich. Anhand der Positions- und Höhendaten lässt sich der Verlauf des ACAS-Events nachvollziehen und dessen Sinnfälligkeit beurteilen. Eine Stärke des ACAS-Monitoring durch mithörende Bodenstationen ist es, ein hoch präzises Höhenprofil für alle Flugzeuge aufzubauen. Senden die Flugzeuge zusätzlich noch ADS-B-Positionsmeldungen aus, so sind auch genaue Positionen mit häufigen Updates vorhanden. Darum beschreibt dieses Kapitel die Beschaffung und die Verarbeitung von Höhen und Positionsdaten im ACAS-Monitoring.

Die Höhen und Positionsdaten sind Teil des Umgebungsmodells `Plane_DB` und werden unabhängig davon, ob ein ACAS-Event eingetreten ist, zu jedem beobachteten Flugzeug im `Plane`-Objekt aufgebaut. Damit lässt sich das Umgebungsmodell auch für andere Untersuchungen heranziehen, etwa um Fehler in den ADS-B-Ausrüstungen zu finden. Hier sollen die Bahndaten aber als Zusatzinformationen zu den ACAS-Events verwendet werden. Wird ein ACAS-Event gefunden, werden die Bahndaten zusammen mit dem RA-relevanten ACAS-Kommunikationen ausgegeben und dargestellt.

7.1 Trennung von vertikalen und horizontalen Spuren

Auf den ersten Blick erscheint es naheliegend, aus den Höhendaten und den Positionsdaten eine 3D-Spur aufzubauen und diese als eine Datenstruktur, eine Liste von Punkten mit Länge, Breite und Höhe abzulegen. Auf den zweiten Blick erkennt man aber, dass besser zwei Datenstrukturen, eine für das Höhenprofil und eine für die horizontale Spur aufgebaut werden sollten:

Denn die DF00-Air-Air-Surveillance-Antworten liefern barometrische Höhen, die ADS-B-DF17-Extended-Squitter liefern horizontale Positionen **und** barometrische Höhen. DF17 werden zweimal pro Sekunde gesendet. DF00 werden als Antworten auf Abfragen dritter ACAS-Systeme gesendet. Da ständig mehrere ACAS-Flugzeuge in der Umgebung sind, werden für gewöhnlich wesentlich mehr DF00 generiert als DF17. Um eine gemeinsame 3D-Spur aufzubauen, kann man Positionsdaten interpolieren, DF00-Höhen ignorieren oder in der 3D-Spur aufwändig zwischen Plots mit und ohne Position unterscheiden. Alle drei Varianten sind unbefriedigend.

Daher wird ein Höhenprofil (Attribut `baro_alt` in Klasse `Plane`) aufgebaut, das Höhen aus allen Quellen (vornehmlich DF00 und DF17) aufnimmt. Daneben wird eine ausschließlich horizontale Spur in Klasse `Track` und deren Helfer `OnePos` (Bild 15 Seite 42) aufgebaut. `Baro_Alt` speichert Zeitpunkte und dazugehörige Höhen und ist wegen des relativ einfachen Aufbaus direkt in der Klasse `Plane` implementiert. Die Klasse `Track` speichert hingegen Zeitpunkte und dazugehörige Positionen, dies ist komplexer und deshalb in einer eigenen Klasse gekapselt.

Diese Aufteilung hat weitere Vorteile und Entsprechungen in der Flugführung: Piloten haben ein Vertical Situation Display und ein Horizontal Situation Display; Lotsen leiten nach einem Radar-

schirm. Der wichtigste Grund ist, dass ACAS vertikale Anweisungen gibt. Um diese RAs und die Reaktion der Flugzeuge genau zu bestimmen, wird vor allem eine hoch präzise Darstellung der Höhenprofile gebraucht. Die horizontale Situation muss nicht mit derselben Genauigkeit beschrieben werden, ist sie nicht vorhanden (weil das Flugzeug kein ADS-B aussendet), können auch schon aus der vertikalen Situation grundlegende Schlüsse gezogen werden.

Effiziente Verwaltung von Höhenprofilen und horizontalen Spuren, Map als sortierte Random Access Liste

Sowohl das Höhenprofil (`baro_alt`) als auch die horizontale Spur (`Track`) sind Zuordnungen von Zeitpunkten zu Plots (Höhe bzw. Positionen). Diese Zuordnung kann effizient mit einer Dictionary-Datenstruktur (in C++ die STL-Map) implementiert werden. Für spätere Analysen müssen allerdings auch Höhen (bzw. Positionen) zu beliebigen Zeitpunkten interpoliert werden können. Zur Interpolation benötigt man den letzten Plot vor und den ersten Plot nach einem gesuchten Zeitpunkt.

Während in anderen (modernerer) Programmiersprachen es heute üblich ist, Hash-Maps [Cormen01] zu benutzen, um Dictionary-Datenstrukturen anzubieten, baut die C++-STL-Map auf binären Bäumen [STL] auf. Die STL-Map lässt sich, im Gegensatz zu Hash-Maps, effizient nach „nächstgelegenen“ Indizes durchsuchen und ist somit perfekt als Index der Höhenprofile und Flugzeugspuren geeignet.

Die Klasse `Track` benutzt die STL-Map, um Anfragen zu einer Position zu einem bestimmten Zeitpunkt zu beantworten. `Track` versucht erst eine gespeicherte empfangene Position zum gefragten Zeitpunkt zu finden. Ist keine vorhanden, sucht `Track` automatisch die nächstliegenden Positionen und interpoliert linear. Das `Plane`-Objekt benutzt `baro_alt` analog.

7.2 Höhen

Bereits im letzten Abschnitt wurde klar, wie Höhendaten zu verarbeiten sind. Die Map `baro_alt` speichert die Zuordnung von Zeitpunkten zu Höhen.

Filterung der Höhen

Da trotz CRC-Check immer wieder falsche Höhenmeldungen empfangen werden, müssen die Höhen gefiltert werden. Eine Höhe wird nur als richtig angenommen, wenn sie zu den vorhergehenden Höhen „passt“. Eine Höhe „passt“, wenn in den letzten 60 s keine Höhe vorhanden war (Neuinitialisierung des Höhenprofils) oder die Vertikalgeschwindigkeit von der letzten Höhe aus unter 30000 ft/min lag. Die erlaubten 30000 ft/min erscheinen im ersten Moment sehr viel, doch die höchste beobachtete Vertikalgeschwindigkeit in einem Event betrug 15.000 ft/min im Steigflug.

Ungleich komplexer als die Verarbeitung der Höhen ist die Verarbeitung der Positionsdaten. Diese wird in den restlichen Abschnitten dieses Kapitels ausführlich dargestellt.

7.3 Vergleich von Mode A/C-Radar, Mode S-Radar und ADS-B-Positionsdaten

Dafür kommen drei Datenquellen für Positionsdaten in Frage:

- Mode A/C-Radardaten empfangen mit Mode A/C-Radar
- Mode S-Radardaten empfangen mit Mode S-Radar
- ADS-B-Daten empfangen mit 1090-Empfangsanlage

Im SATT-Projekt sollen vor allem ADS-B-Daten verwendet werden, weil diese wegen ihrer Auflösung und Wiederholrate die größte Genauigkeit erreichen und vor allem, weil sie bei der Beobachtung von ACAS-Kommunikationen mit empfangen werden. Dieser Abschnitt vergleicht diese drei Datenquellen und begründet die Entscheidung für ADS-B. Zum Vergleich wird die Landung eines beliebigen Fluges in Frankfurt/Main herangezogen. Um die Positionsdaten anzuzeigen, wurde die Software ADS-B-Display der Firma Thales benutzt. Zusätzlich wurde noch selbst die Software radar2asx entwickelt, die die Radardaten in ein vom ADSB-Display darstellbares Format übersetzt.

7.3.1 Radardaten der DFS

Die DFS hat dem IfEV im Rahmen des SATT-Projektes vereinzelt und auszugsweise Radardaten (Asterix CAT-62) zur Verfügung gestellt. Die Datensätze enthalten sowohl Mode A/C- als auch Mode S-Daten. Die Mode A/C-Daten wurden mit im Flugsicherungsbetrieb befindlichen Mode A/C-Radaranlagen aufgenommen und wurden einer Spurvereinigung, Spurbildung und Filterung unterzogen. Die Mode S-Daten wurden dagegen von wenigen vereinzelt Mode S-Radar-Anlagen aufgenommen und keiner Spurvereinigung unterzogen. D. h. in den Aufzeichnungen existiert für jedes Target (vom Radar erfasstes Flugzeug) eine Mode A/C-Spur, für Targets, die von einem Mode S-Radar erfasst wurden, existiert zusätzlich noch eine Mode S-Spur.

Aufbau der Datensätze

Einen Ausschnitt aus einer Radaraufzeichnung zeigt Listing 5. Die Datensätze sind zeilenweise in einer derartigen Textdatei gespeichert.

```
00:09:39, 271,7777,N485641.86,E114326.67, 94900,209.1, 5.0,TT,,,,,0
00:09:39, 1748,0000,N495438.93,E083347.52, 0,203.9,45.7,,,,,0
00:09:40, 2400,0000,N500033.07,E083913.35, 600,0.0, 0.0,,THALES,3c9a61,,1
00:09:40, 2409,0000,N503119.98,E063235.97, 22000,293.4,394.6,,BCS964,44918a,,1
00:09:40, 2403,0000,N481546.58,E090424.00, 37000,134.0,457.4,,ELY318,7380a2,,1
00:09:40, 877,0000,N532835.65,E095822.03, 0,130.6,50.0,,,,,0
```

Listing 5: Ausschnitt aus den Radardaten

Tabelle 5 zeigt den Aufbau eines Radardatensatzes. „Nr.“ ist die Spalte im Datensatz, „TH-Nr“ ist die Spalte im vom radar2asx generierten Datensatz.

Nr.	Name	Bedeutung	TH-Nr:
1	Time	Uhrzeit	13
2	Track ID	Identifikation der Spur	16
3	SSR-Code	Squawk, wird bei Mode S-Targets nicht angegeben	90
4	Lat	geographische Breite in Grad, Minuten, Sekunden	14
5	Lon	geographische Länge in Grad, Minuten, Sekunden	15
6	Alt in ft	Druckhöhe in ft, Test Targets haben unmöglich große Alt	17
7	Heading	Winkel gegen geographisch Nord, rechtsdrehend	36
8	Speed	Geschwindigkeit über Grund in kt	31
9	Flightplan Callsign	Rufzeichen, TT = Test Target	
10	Mode S Callsign	Mode S- Rufzeichen	39
11	Mode S AA	Mode S-Adresse	
12	Aircraft Type	Flugzeug Typ	
13	ADSB Marc	ADS-B-Markierung: 1 hat ADS-B, 0 hat kein ADS-B	

Tabelle 5: Aufbau der DFS-Radardaten

Beispiel Spuren

In Bild 24 wird die Mode A/C-Spur des beliebig gewählten Landeanflugs auf Frankfurt gezeigt. Bild 25 zeigt die gleiche Spur, nur diesmal mit einem Mode S-Radar aufgezeichnet.

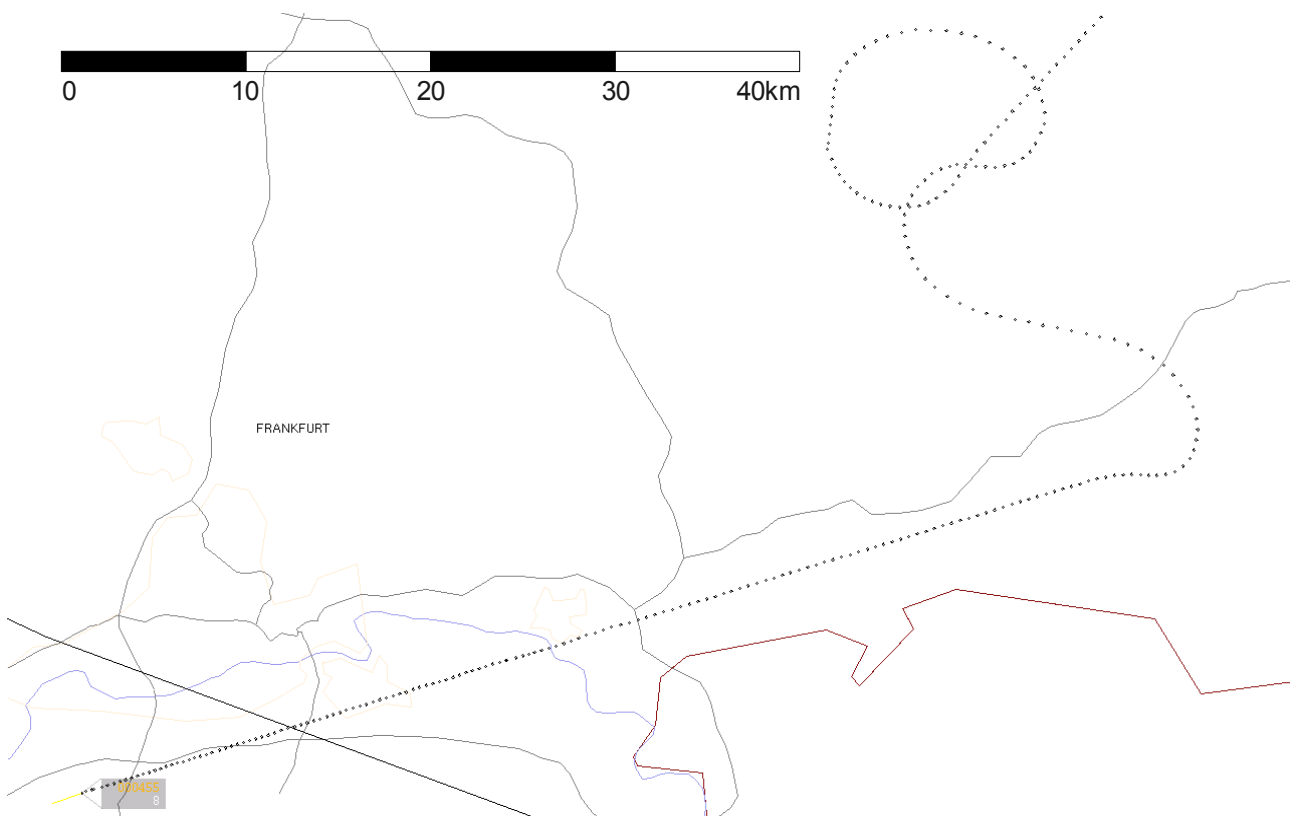


Bild 24: Mode A/C-Spur beim Landeanflug auf Frankfurt

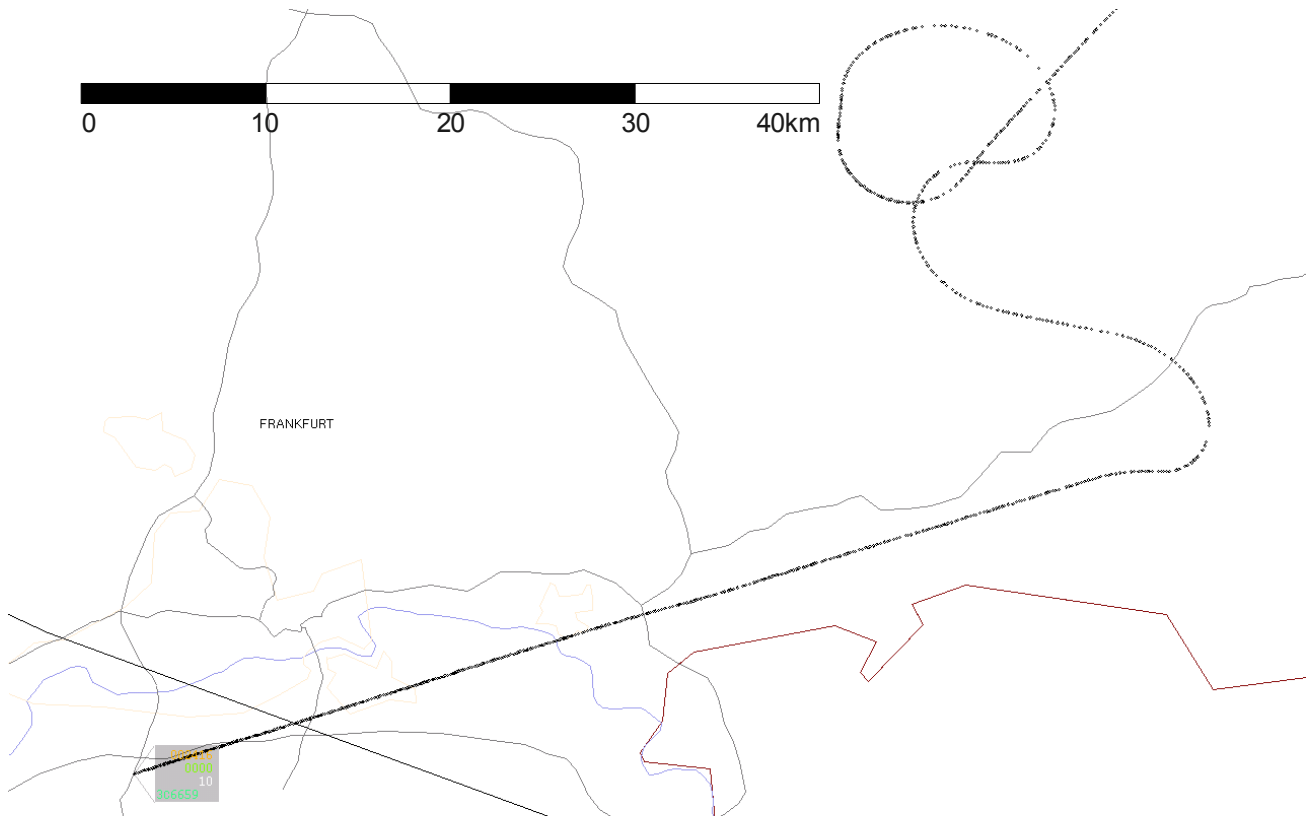


Bild 25: Mode S-Spur beim Landeanflug auf Frankfurt

7.3.2 ADS-B

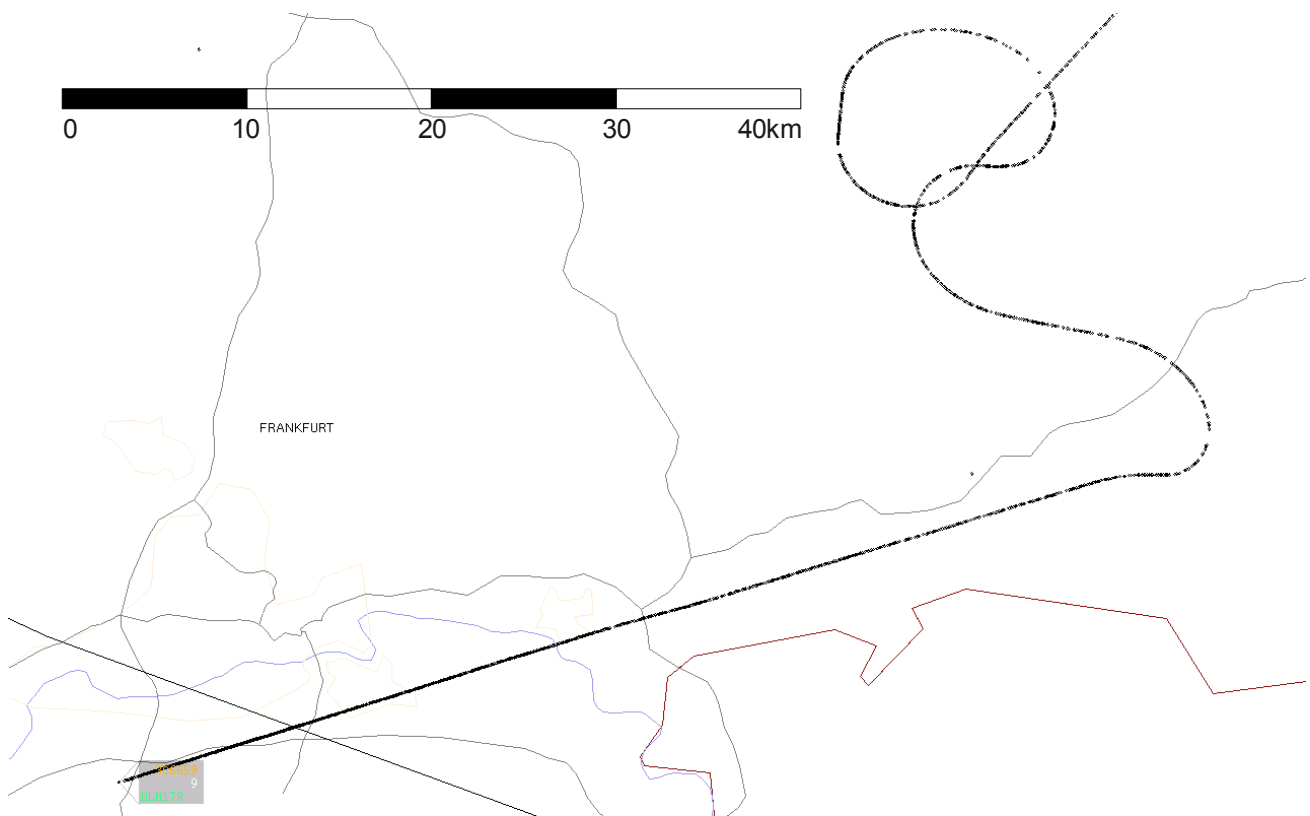


Bild 26: ADS-B-Spur beim Landeanflug auf Frankfurt

Bild 26 zeigt die ADS-B-Spur des Beispiel-Anfluges.

Mit ADS-B ausgerüstete Flugzeuge senden nominell zweimal pro Sekunde ihre Position als DF17 aus. Je nach Entfernung des Senders empfängt man tatsächlich zwischen zwei Positionsmeldungen pro Sekunde und einer alle 20 s.

Noch sind nicht alle Flugzeuge mit ADS-B ausgerüstet, wodurch sich ADS-B zur Zeit nicht eignet, um den kompletten Flugverkehr zu beobachten. Die Stärke von ADS-B liegt vielmehr in der einfachen Verfügbarkeit. Eine Mode S-Bodenstation zum Mithören von ADS-B-Positionen ist im Vergleich zu einem Radar relativ einfach und preisgünstig aufzubauen. Des Weiteren haben ADS-B-Positionsdaten bei gutem Empfang eine deutlich höhere Erneuerungsrate als Radardaten.

7.3.3 Genauigkeit der Positionsmeldungen

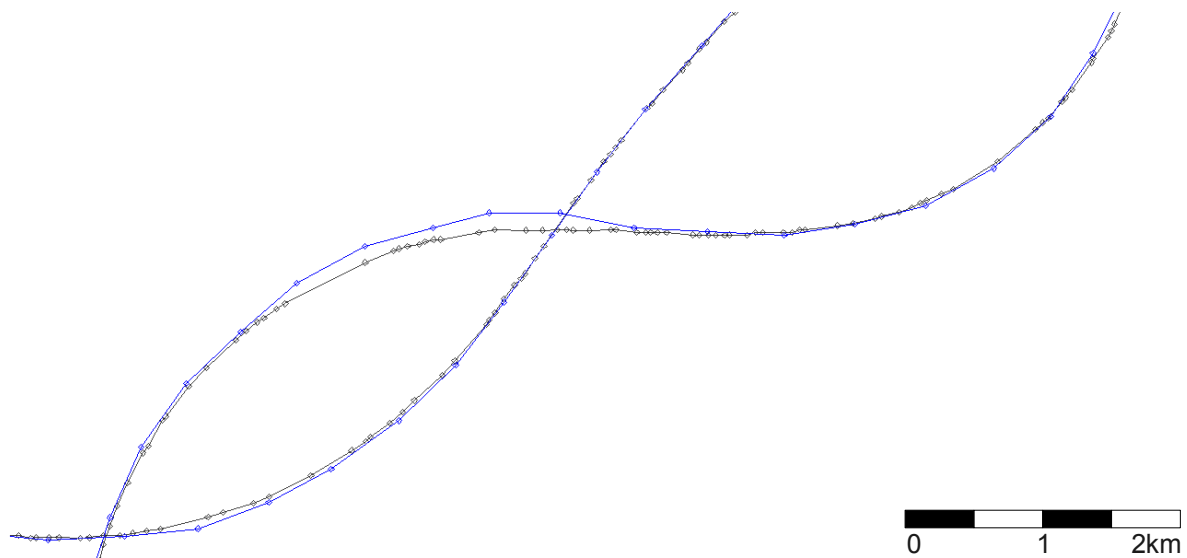


Bild 27: Spuren: ADS-B (grau) <-> Mode A/C (blau), Größter Abstand: 150 m

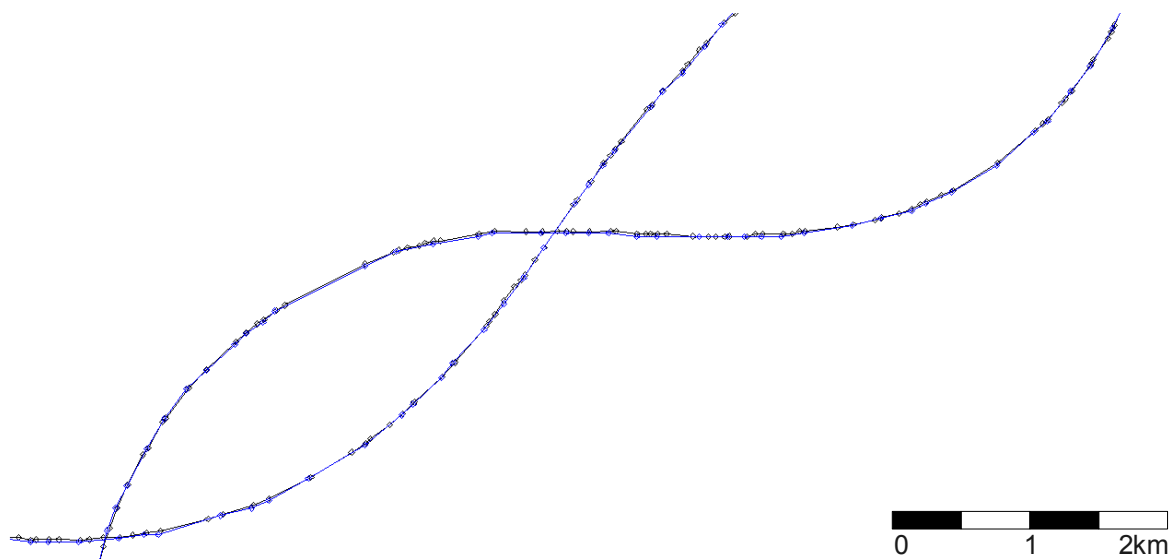


Bild 28: Spuren: ADS-B (grau) <-> Mode S (blau), beide Spuren sind fast deckungsgleich

Bild 27 und 28 stellen die ADS-B-Spur und die Mode A/C- bzw. Mode S-Spuren nebeneinander dar.

Bild 28 zeigt, dass ADS-B und Mode S kaum voneinander abweichen. Beide Spuren erscheinen daher korrekt. Die Mode A/C-Spuren (Bild 27) weichen teilweise bis zu 150 m von der ADS-B-Spur ab. Diese Abweichungen können auf das Tracking der Mode A/C-Spur zurückgeführt werden.

7.3.4 Häufigkeit der Positionsmeldungen

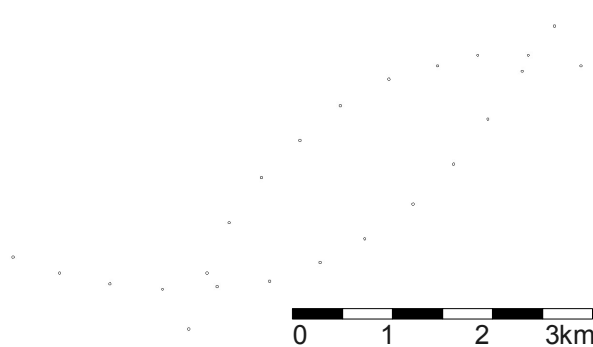


Bild 29: Ausschnitt aus Mode A/C-Spur

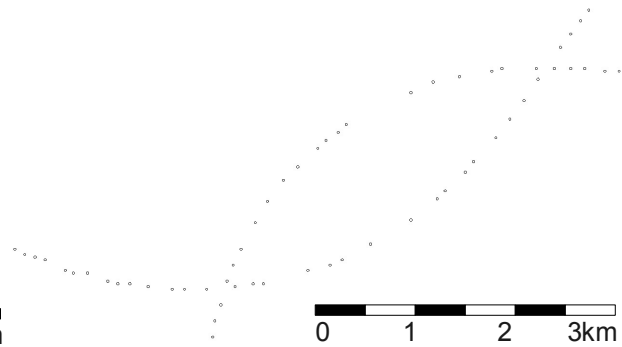


Bild 30: Ausschnitt aus Mode S-Spur

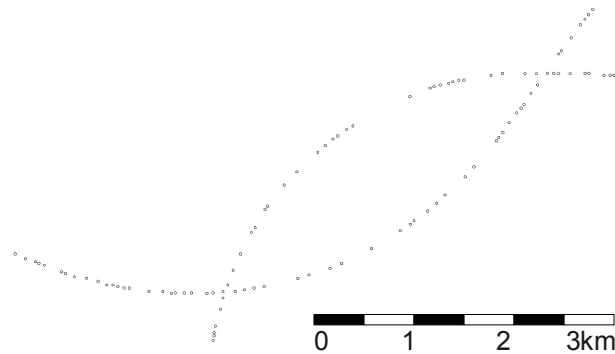


Bild 31: Ausschnitt aus ADS-B-Spur

Bild 29 bis 31 zeigen Ausschnitte aus Spuren. In welchen Zeitabständen sich ein Flugzeug meldet, gibt Tabelle 6 an.

	Mittlerer Abstand $\pm \sigma/2$	Maximaler Abstand
Mode A/C	4,8 s \pm 0,4 s	5,0 s
Mode S	2,1 s \pm 1,0 s	6,0 s
ADS-B	1,1 s \pm 0,9 s	5,9 s

Tabelle 6: Mittlere Zeit zwischen Positionsmeldungen in Bild 29 bis 31

7.4 ADS-B-Decodierung

Dieser Abschnitt beschäftigt sich nur mit der Decodierung von horizontalen Positionsdaten aus ADS-B-Meldungen. Mit ADS-B ausgerüstete Flugzeuge übertragen ihre Position in DF17-Squittern. DF17 Nachrichten werden von der Funktion `interpret_DF17()` decodiert. Diese ruft dann die Funktion `interpret_me()` auf, die für das ME-Feld zuständig ist. Handelt es sich um eine Positionsmeldung, so wird `interpret_airborne_pos()` aufgerufen.

CPR-Decodierung

ADS-B benutzt die so genannte CPR-Codierung (Compact Position Reporting). Dabei werden die geographische Breite und Länge mit jeweils 19 Bit codiert, von denen nur die letzten 17 übertragen werden. Zusammen mit der 12-Bit-Höhe werden Länge und Breite im 56-Bit-ME-Feld einer DF17 übertragen. Weil die ersten 2 Bit der Länge und Breite weggelassen wurden, ist die Position, die aus einer einzigen DF17 hervorgeht, nicht eindeutig. Diese nicht-triviale Codierung ist zusammen mit dem gesamten ADS-B im ICAO Manual on Mode S Specific Services [ICAO9688], in ICAO: „Technical Provisions for Mode S Services and Extended Squitter“ [ICAO9871] und in RTCA: „Minimum Operational Performance Standards for 1090 MHz Extended Squitter Automatic Dependent Surveillance – Broadcast (ADS-B) and Traffic Information Services – Broadcast (TIS-B)“ [RTCA DO260A] erklärt.

Local Decoding

Ist die ungefähre Position eines Flugzeuges auf 666 km bekannt (z. B. durch eine alte bekannte Position oder aus der Position und Reichweite der Bodenstation), so lässt sich die aktuelle Position eindeutig aus einer einzigen DF17-Positionsmeldung bestimmen. Dies wird als Local Decoding bezeichnet.

Global Decoding

Um dennoch nur mit Hilfe von DF17-Positionsmeldungen eine eindeutige Position zu beschreiben, benutzt CPR verschiedene Codierungen in geraden und ungeraden Sekunden, die als „gerade“ bzw. „ungerade“ Codierung bezeichnet werden. Welche Codierung gerade gesendet wird, wird in der DF17 durch ein Bit angezeigt. Werden eine gerade und eine ungerade DF17 (die weniger als 10 s auseinander lagen) verknüpft, so kann die eindeutige Position des Flugzeuges bestimmt werden. Dies wird als Global Decoding bezeichnet.

7.4.1 Ablauf der CPR-Decodierung in der ACAS-Monitor-Software

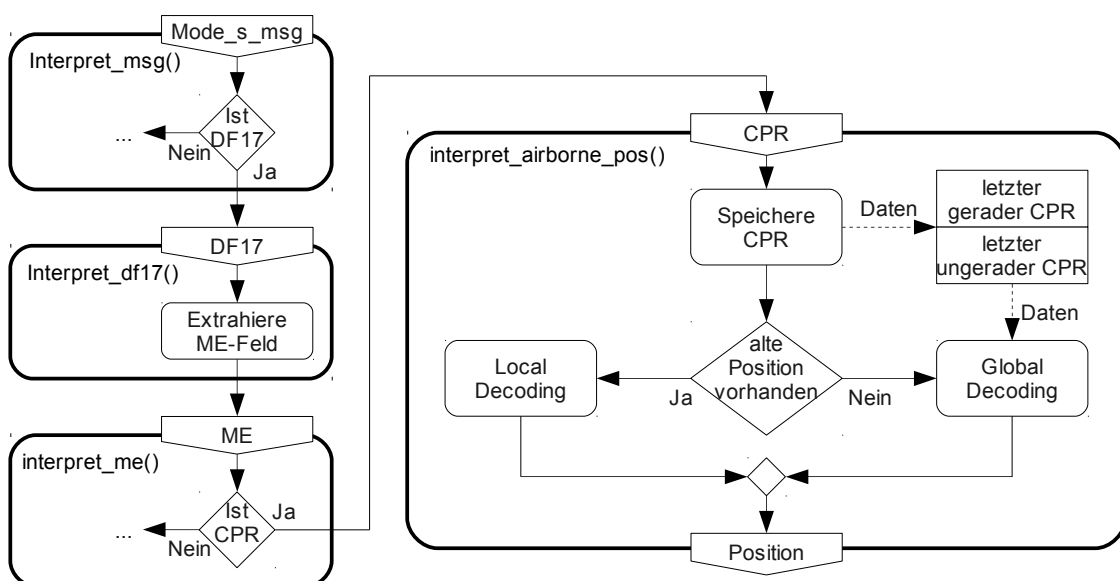


Bild 32: CPR-Decodierung in der ACAS-Monitor-Software

Die Funktion `interpret_airborne_pos()` steuert die CPR-Decodierung. `interpret_airborne_pos()` erhält die aktuelle CPR-Positionsmeldung und das betreffende Plane-Objekt als Parameter. Für das Global Decoding enthält die Klasse Plane Speicher (Attribute) für die jeweils letzte gerade und die letzte ungerade CPR-Positionsmeldung. `interpret_airborne_pos()` schreibt die aktuelle CPR-Positionsmeldung in das entsprechende Attribut von Plane.

Ist eine aktuelle Position (neuer als 10 s, gesichert durch Spurbildung nach Abschnitt 7.5) vorhanden, wird Local Decoding benutzt. Die resultierende eindeutige Position wird mittels `Track::maybe_add_pos()` der Spurbildung zugeführt und gegebenenfalls zur Spur des Flugzeuges hinzugefügt.

Ist keine aktuelle Position vorhanden, so muss auf Global Decoding zurückgegriffen werden. Dabei wird die passende gespeicherte CPR-Meldung und die aktuelle CPR-Meldung zu einer eindeutigen Position verknüpft. Es resultieren zwei eindeutige Positionen: Eine zur aktuellen CPR-Meldung und eine zur alten CPR-Meldung. Beide werden mittels `Track::maybe_add_pos()` der Spurbildung zugeführt und gegebenenfalls zur Spur des Flugzeuges hinzugefügt.

7.5 ADS-B-Filterung, Spurinitiierung und Spurbildung

Die decodierten CPR-Meldungen enthalten immer wieder falsche Positionen. Dabei werden oft einzelne oder auch mehrere sinnlose Positionen nacheinander empfangen, die sogar Spuren ergeben. Um das zu verhindern, hat sich die folgende Art der Spurinitiierung und -bildung (Bild 33) bewährt und bisher als ausreichend erwiesen. Die Spurinitiierung und -bildung übernimmt die Klasse Track selbst mit ihrer Funktion `maybe_add_pos()`.

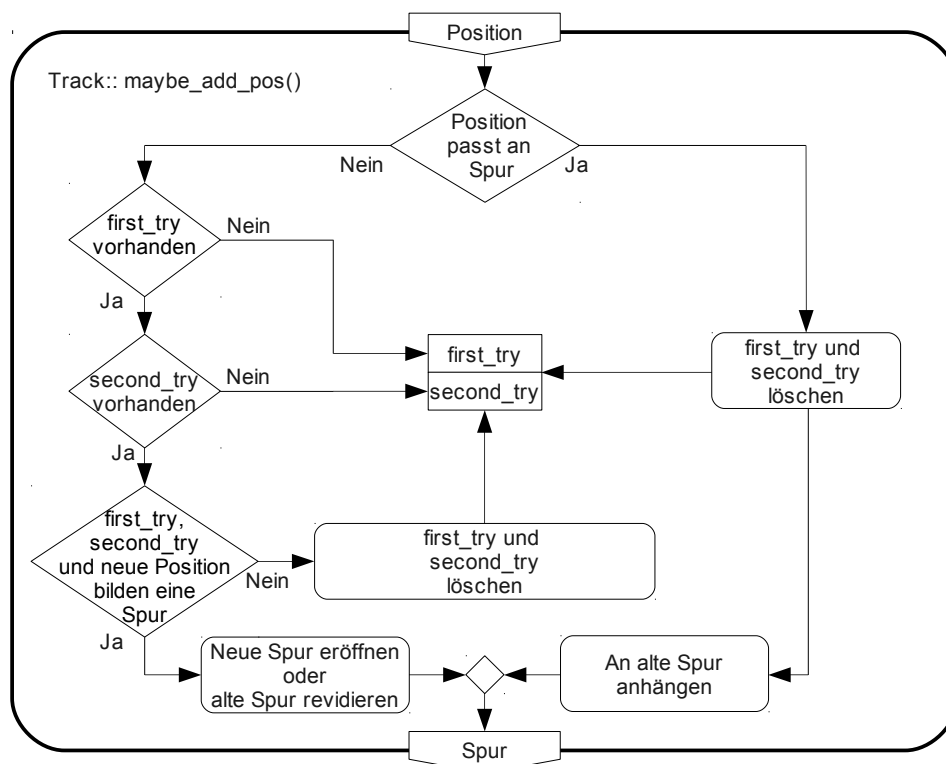


Bild 33: Spurinitiierung und -bildung

7.5.1 Positionen „passen zusammen“ – `can_be_neighbors_in_track()`

In `Track::maybe_add_pos()` muss mehrfach geprüft werden, ob zwei Positionen (OnePos a und OnePos b) zusammen eine Spur bilden können. Dazu wird in `can_be_neighbors_in_track()` der maximale Abstand zwischen diesen Positionen geprüft. Außerdem werden Positionen nur dann als benachbart angesehen, wenn sie maximal 10 min auseinander liegen.

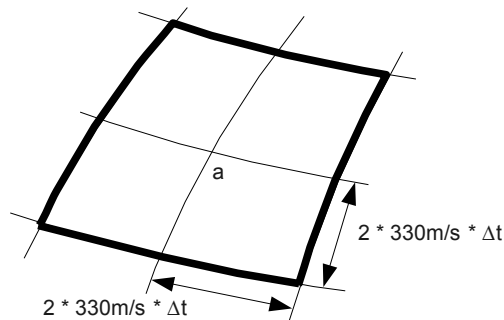


Bild 34: Maximal zugelassener Abstand zwischen decodierten Positionen

Um den Berechnungsaufwand gering zu halten, wird der Abstand entlang der geographischen Länge und Breite getrennt geprüft. Es wird davon ausgegangen, dass sich ein beobachtetes Flugzeug maximal mit Schallgeschwindigkeit (330 m/s) in irgendeine Richtung fortbewegt haben kann. Um Jitter Effekte bei der Datierung der DF17 im Empfänger auszugleichen, wurde der Sicherheitsfaktor 2 eingeführt. Der maximale Abstand in eine Richtung beträgt also $2 \cdot 330 \text{ m/s} \cdot \Delta t$ ($\Delta t \sim$ Differenz der Zeitstempel der Positionsmeldungen). Sind die Positionen a und b nicht mehr als 10 min und nicht mehr als $2 \cdot 330 \text{ m/s} \cdot \Delta t$ in Länge und Breite auseinander, so gibt `can_be_neighbors_in_track(a, b)` `true` zurück. Im Weiteren wird dies „a passt zu b“ genannt.

7.5.2 `Track::maybe_add_pos()`

`Track::maybe_add_pos()` (Bild 33, Seite 65) entscheidet, ob eine neue Position zu einer Spur hinzugefügt wird. Die Klasse `Track` speichert neben der eigentlichen gesicherten Spur auch zwei Positionen, die den möglichen Anfang einer neuen Spur darstellen – `first_try` und `second_try`. Zunächst prüft `maybe_add_pos()` durch `can_be_neighbors_in_track()`, ob die neue Position zur letzten Position in der Spur passt. Ist dies der Fall, so wird die neue Position an den Anfang der Spur gesetzt und `first_try` und `second_try` werden geleert.

Passt die neue Position nicht zur alten Spur (weil z. B. noch keine Spur vorhanden ist), so wird die neue Position in `first_try` abgelegt. Ist `first_try` bereits belegt, dann wird die neue Position in `second_try` abgelegt. Ist auch `second_try` belegt, wird versucht aus den drei Positionen eine neue Spur zu bilden.

Um eine neue Spur zu bilden, müssen `first_try` und `second_try` sowie `second_try` und die neue Position zusammen passen. Ist dies der Fall, so wird eine neue Spur mit diesen drei Positionen angelegt, unabhängig davon ob bereits eine alte andere Spur vorhanden ist. Die Speicher für `first_try` und `second_try` werden geleert. Passen die drei Positionen nicht zusammen, wird die nicht passende entfernt, im Zweifel `first_try`.

8 Vernetzung von ACAS-Monitor-Stationen

In den vorangegangenen Kapiteln wurde gezeigt, wie ACAS-Monitoring mit einer Station, bestehend aus einer 1030MHz/1090MHz-Empfangsanlage und einem Auswertungsrechner, durchgeführt werden kann. Mit einem solchen Einzelsystem ist ACAS-Monitoring nur auf ein Gebiet mit wenigen 100 km Radius begrenzt. Die Krümmung der Erde, Gebirge und Überlagerungen von gleichzeitig gesendeten Nachrichten begrenzen die Reichweite einer Station.

Darum ist es sinnvoll, mehrere Empfangs-Stationen (im Folgenden einfach Stationen genannt) über ein großes Gebiet verteilt aufzustellen und in geeigneter Weise zu einem ACAS-Monitor-Netz zu verbinden. Dabei stellt sich die Frage nach der Architektur des Netzwerkes, der Aufstellung und Anbindung der Stationen und der Implementierung der Analysesoftware. In diesem Kapitel werden diese Aspekte diskutiert und gezeigt, wie sie im ACAS-Monitor-Netz der TU Braunschweig umgesetzt wurden.

2008 erhielt das Institut für Eisenbahnwesen und Verkehrssicherung die Möglichkeit, mehrere Mode S-Empfangsstationen zu beschaffen und aufzustellen. Diese sollten bei befreundeten Unternehmen und Forschungseinrichtungen kostenneutral aufgestellt werden. Als Gegenleistung sollten die Partner die ADS-B-Daten der bei ihnen aufgestellten Station erhalten.

Im folgenden Abschnitt 8.1 sollen zunächst mögliche Architekturen eines ACAS-Monitor-Netzes erarbeitet und verglichen werden.

8.1 Mögliche Architekturen des ACAS-Monitor-Netzes

ACAS-Monitoring soll in einem möglichst großen Gebiet mit möglichst wenig Aufwand betrieben werden. Aufwändig ist hier aber vor allem die Installation von Empfangsstationen. Jede Station muss gekauft werden und braucht einen geeigneten Antennen-Standort, an dem Strom und Netzanbindung verfügbar sind. Der Blick zum Horizont muss frei sein und es dürfen keine Störungen, z. B. durch Mobilfunkmasten in unmittelbarer Nähe entstehen. Solche Standorte zu finden ist aufwändig und kostenintensiv.

Es müssen also Empfangsstationen über das zu bedeckende Gebiet verteilt werden. Die Empfangsstationen müssen zu einem ACAS-Monitor-Netzwerk verbunden werden. Die Frage ist jetzt, wie werden die Module des ACAS-Monitor-Prozesses in diesem Netz verteilt, um maximale Beobachtungskapazität bei minimalem Aufwand zu erreichen.

Die 1030 MHz und 1090 MHz Signale lassen sich nur schwer analog von der Antenne zu einem mehrere 100 km entfernten Auswertesystem übertragen. Deshalb werden bei allen Ansätzen die Empfangsdaten am Antennenstandort digitalisiert und über digitale Netze übertragen. Da heute die meisten großen Netzwerke auf dem Internetprotokoll (IP) basieren, wird von vornherein IP als Vermittlungsprotokoll gewählt. Außerdem wird ein Wide-Area-Netzwerk (WAN) auf IP Basis vorausgesetzt, über das die Stationen verbunden sind. Dieses WAN wird das Internet bilden, kann aber auch durch ein anderes Netzwerk realisiert werden.

Um das ACAS-Monitor-Netz aufzubauen, bieten sich folgende Architekturen an:

1. Empfangsanlage und ACAS-Monitor befinden sich im Feld. Die ACAS-Monitor-Prozesse arbeiten unabhängig voneinander. Die fertigen Event-Reports werden zentral gesammelt.
2. Empfangsanlagen übermitteln (fast) alle Empfangsdaten an ein zentrales Analysesystem.
3. Zu jeder Empfangsanlage gibt es einen eigenen ACAS-Monitor-Prozess. Diese ACAS-Monitor-Prozesse arbeiten synchron als eine verteilte Anwendung in einem Peer-to-Peer Netz von Gleichrangigen. Die ACAS-Monitor-Prozesse tauschen nur Informationen zu ACAS-Events aus, insbesondere dann, wenn mehrere Stationen Informationen zu einem ACAS-Event beitragen können.

8.1.1 Architektur 1: Autonome ACAS-Monitor-Prozesse in den Empfangsstationen

Bei diesem Ansatz wird das nicht-verteilte ACAS-Monitoring-System einfach dupliziert. Die Empfangsanlage, der Aufzeichnungsprozess und der Auswerteprozess sind in den Stationen. Fertige Event-Reports werden an ein zentrales System übermittelt und dort gesammelt.

Wird ein Event von mehreren Stationen empfangen, so entstehen mehrere Event-Reports. Diese Reports müssen entweder manuell oder automatisch vereinigt werden. Bei einem großen Netz ist die manuelle Vereinigung nicht mehr effektiv und eine automatische Vereinigung von Events muss durchgeführt werden.

Dieser Ansatz skaliert sehr gut, das heißt, es kann leicht ein experimentelles Netz mit wenigen Stationen zu einem großen Netz mit vielen Stationen ausgebaut werden, da der rechenintensive Auswerteprozess isoliert in jeder Station ausgeführt wird und damit von selbst parallelisiert ist. Nachteilig ist, dass sich sehr viel Intelligenz in den Stationen befindet, das steigert den administrativen Aufwand.

Ein beträchtlicher Aufwand ergibt sich, wenn einzelne oder alle Auswertungen wiederholt werden sollen, weil z. B. die Auswertesoftware verbessert wurde. Dazu müssten die Mode S-Daten archiviert werden, entweder lokal in den Stationen (was z. B. einen halbjährlichen Festplattenwechsel erfordert) oder zentral, was eine Breitbandanbindung der Stationen und ein zentrales Speichersystem erfordert. Ein zentrales Speichersystem verringert die Skalierbarkeit, da es bei einem großen Netz zum Flaschenhals werden kann. Weiterhin braucht man eine zweite zentrale Auswerteeinheit, um damit die Auswertungen wiederholen zu können.

Jedoch macht dieser Ansatz nur wenig Nutzen von Mehrfachbedeckungen zwischen Stationen: Werden Events von mehreren Stationen beobachtet, könnten diese gemeinsam zum Event beitragen und so bei schlechtem Empfang in mehreren Stationen eine gemeinsame gute Event-Beschreibung erzeugen. Bei Architektur 1 wird Mehrfachbedeckung jedoch erst nach dem Erkennen und Generieren der einzelnen Event-Reports durch Vereinigung zusammengehöriger Events genutzt. Zum Erkennen der ACAS-relevanten Nachrichten ist jedoch eine umfangreiche Wissensbasis nötig: Die Flugzeuge müssen durch Squitter der Bodenstation bekannt sein und die Squawks der Flugzeuge müssen den Bodenstationen durch DF05 bekannt werden. Es kann durchaus sein, dass diese Informationen in einer benachbarten Station vorliegen, aber nicht in der Station, die gerade diese Information braucht. Dieser Nachteil kommt vor allem bei grobmaschigen Netzen mit wenig Stationen, deren Empfangsbereiche sich aber immer noch teilweise überdecken, zum Tragen.

8.1.2 Architektur 2: Empfangsanlagen im Feld senden an zentrale Auswertung

Bei diesem Ansatz senden die Empfangsanlagen ihre Empfangsdaten an ein zentrales System. Das zentrale System vereinigt die Empfangsdaten im zeitlichen Reißverschlussverfahren zu einem kombinierten Datenstrom, der dem ACAS-Monitor als eine Datenquelle übergeben wird.

Die Stationen sind relativ einfach. Ihre Aufgaben sind

- Empfangsdaten von den Empfangsanlagen entgegennehmen.
- Datierung der Nachrichten vervollständigen: die Thales SPUs datieren nur mit Sekunden seit Mitternacht, ein Datum muss noch hinzugefügt werden.
- Die Empfangsdaten zwischenspeichern. Dies ist hilfreich, um Engpässe oder Ausfälle der Internetanbindung oder des zentralen Servers zu überbrücken. Das kommt dem Aufbau als experimentelles Netz sehr entgegen.
- Die Empfangsdaten an den zentralen Server übermitteln. Hierfür ist ein Breitbandanschluss erforderlich.

Die Intelligenz des ACAS-Monitor-Netzes konzentriert sich im zentralen Analyse-System, in dem die Empfangsdaten zusammengeführt und gemeinsam analysiert werden. Dieser zentralistische Aufbau verringert den Administrations- und Entwicklungsaufwand. Beim Zusammenführen der vorsortierten Empfangsdatenströme anhand der Empfangszeiten handelt es sich um einen sog. Merging Prozess, der lediglich eine Zeitkomplexität von $O(n)$ (bei n zu sortierenden Nachrichten) hat – ein Sortieren von ungeordneten Datenströmen hätte die Laufzeitkomplexität $O(n \log n)$.

Der eigentliche Vorteil ist die maximale Ausnutzung von Mehrfachbedeckung. Da alle Stationen zu einer gemeinsamen Wissensbasis beitragen, sind alle Informationen zentral verfügbar, egal, wo sie empfangen werden. So lassen sich Verknüpfungen zwischen Mode S-Nachrichten von verschiedenen Stationen herstellen.

Leider verringert sich auch die Skalierbarkeit (die Fähigkeit, durch Hinzufügen neuer Komponenten das Netz zu vergrößern), da die Auswertung durch das Merging nicht parallel in allen Stationen stattfindet. Dies kann ausgeglichen werden, indem für jede Region eine Zentrale eingerichtet wird, Stationen an den Grenzen dieser Regionen senden dann an zwei Zentralen.

Ein weiterer Vorteil ist, dass Analysesystem und Speichersystem am gleichen Ort sind. Damit lassen sich leicht alte Aufzeichnungen mit einer überarbeiteten Software analysieren. Die Software für die automatische umgehende Analyse neuer Daten ist zum Großteil identisch mit der Software, die zur nachträglichen, erneuten Analyse von Aufzeichnungen verwendet wird. Das kommt einem experimentellen System sehr entgegen.

8.1.3 Architektur 3: Verteilter Peer-to-Peer ACAS-Monitor

Diese Architektur versucht beide oben angesprochenen Systeme zu kombinieren. Für jede Empfangsanlage existiert ein eigener ACAS-Monitor-Prozess. Diese Prozesse können in den jeweiligen Stationen ablaufen und so Datenübertragungsrate sparen, sie können gemeinsam auf einem zentralen Rechner oder Cluster ablaufen oder als Gruppen von Prozessen auf Rechenzentren verteilt werden.

Wichtig ist, dass zu jeder Station ein Auswerte-Prozess existiert. Alle Prozesse laufen synchron, was recht leicht erreicht werden kann, wenn die Prozesse in Realzeit laufen. Die ACAS-Monitor-Prozesse propagieren **nur** wichtige Informationen zu den Nachbarprozessen. Damit benötigen die Stationen keine Breitbandanbindung.

Der Aufwand, um diesen verteilten Algorithmus zu implementieren ist jedoch wesentlich höher als der Aufwand für Architektur 2. Damit scheidet der Algorithmus von vornherein aus und soll auch nicht weiter diskutiert werden.

8.1.4 Wahl der Architektur

Zum Aufbau des Netzes wurde Architektur 2 ausgewählt, weil:

- Obwohl Architektur 1 einfach nur eine Vervielfältigung des ursprünglichen „Ein-Standort-ACAS-Monitors“ ist, kommt auch Architektur 2 mit überschaubaren Softwareänderungen aus: Die Mode S-Daten müssen an das Zentralsystem weitergeleitet werden, um dort automatisch vereinigt und analysiert zu werden. Die Abläufe im Zentralsystem sind zu überwachen und zu steuern.
- Aus vorhandener Mehrfachüberdeckung wird maximaler Nutzen gezogen und so werden in einem Netz mit wenigen Stationen die empfangenen Nachrichten optimal ausgenutzt.
- Die Speicherung der Daten und deren Analyse sowie die eigentliche Softwareentwicklung erfolgen zentral. Die Stationen im Feld müssen nur die Daten weiterleiten, sind somit einfacher als bei den anderen Architekturen aufgebaut und auf die Stationen muss weniger oft zugegriffen werden.
- Die Analyse kann asynchron erfolgen. D. h. die Empfangsdaten werden in den Stationen aufgezeichnet und übertragen, sobald die Internetverbindung und das Zentralsystem bereit sind. Hat das Zentralsystem alle Aufzeichnungen vorliegen, so kann es deren Analyse durchführen.

8.2 Standorte der Bodenstationen

Folgende Überlegungen ergeben sich offensichtlich: Soll ein Gebiet mit einem ACAS-Monitor-Netz abgedeckt werden, müssen genügend Stationen so über das zu bedeckende Gebiet verteilt werden, dass möglichst keine unbedeckten Regionen entstehen. Die Stationen können nicht gleichmäßig verteilt werden, sondern die geografischen Gegebenheiten, wie abschattende Gebirge, sind zu beachten.

Nicht offensichtlich und damit schwerer zu berücksichtigen sind die „weichen“ Faktoren der Standortwahl. Am Aufstellungsort muss ein Stromanschluss und Netzwerkanbindung vorhanden sein. Die Anlage muss trocken stehen. Die Temperaturen am Standort der Empfangsanlage dürfen nicht zu stark schwanken [Thales06]. Die Empfangsanlagen dürfen nicht durch andere Sendeanlagen gestört werden: Aus nächster Nähe einstrahlende SSR-Radare können den Empfänger sogar zerstören, nahe GSM-Mobilfunkmasten können den Vorverstärker oder den Eingangsverstärker übersteuern.

Weiterhin ist es empfehlenswert, Stationen an „Brennpunkten“ aufzustellen: In Gegenden mit hohem Luftverkehrsaufkommen wird auch auf den Mode S-Kanälen viel kommuniziert. Bei sehr hoher Mode S-Funkfeldlast kommt es ständig zu Paket-Überlagerungen im Kanal. Trifft während der Demodulation einer Nachricht eine andere mit gleichem oder stärkerem Pegel ein, so kann keine der Nachrichten demoduliert werden. Wird hingegen eine starke Nachricht von einer schwachen gestört, so kann unter Umständen die starke Nachricht noch fertig demoduliert werden. Auf diesem Weg setzen sich Nachrichten mit starkem Pegel durch. Nachrichten mit starkem Pegel stammen für gewöhnlich von nahen Flugzeugen. So führt eine erhöhte Funkfeldlast zu einer verringerten Empfangswahrscheinlichkeit weit entfernter Flugzeuge. Daher sollten Verkehrsbrennpunkte mit eigenen Stationen versehen werden, um deren Nachrichten mit starkem Pegel zu empfangen und vollständig zu demodulieren.

8.2.1 Standorte der Stationen der TU Braunschweig

Das Institut erhielt 2008 die Möglichkeit, fünf Stationen zu beschaffen. Nach den vorhergehenden Betrachtungen zur Stationsreichweite (Abschnitt 4.4) wird klar, dass sich damit nicht der gesamte deutsche Luftraum abdecken lässt. Ziel des Institutes war es daher, zumindest wesentliche Teile des deutschen Luftraums abzudecken. Weiterhin war das Institut nicht in der Lage, hohe Mietkosten oder Anschlussgebühren an den Aufstellungsorten zu bezahlen, das Institut konnte lediglich die von den Stationen aufgezeichneten Daten anbieten. Deshalb konnten die Stationen nur bei befreundeten Institutionen aufgestellt werden, die Interesse an den Untersuchungsergebnissen oder an den Mode S- bzw. ADS-B-Daten hatten.

Es konnten vier Standorte für fünf Stationen gefunden werden: Doppelstation in Braunschweig auf DLR-Turm, Flughafen Frankfurt, Korntal bei Stuttgart und DLR-Oberpfaffenhofen.

8.2.2 Standort 1: Doppelstation Braunschweig DLR-Turm

Die bisherige Antenne auf dem Dach des Institutes überragt Wälder in südwestlicher und östlicher Richtung nicht, deshalb ist ihre Reichweite unnötig beschränkt. Da Braunschweig inmitten der norddeutschen Tiefebene liegt, lassen sich aber mit einer genügend hohen Antenne große Teile Norddeutschlands beobachten.



Bild 35: Flughafen Braunschweig-Wolfsburg mit Stationsstandorten [OSM]

Ca. 1 km vom Institut entfernt auf dem Gelände des DLR-Braunschweig befindet sich ein geeigneter 30 m hoher Stahlurm (N 52° 18.935' E 10° 33.836', siehe Bild 35). Das DLR-Institut für Flugführung (FL), dem der Turm gehört, war bereit, mit dem IfEV eine Kooperation einzugehen und auf dem Turm eine Mode S-Monitor-Station aufzunehmen. Dazu wurde ein Kooperationsvertrag ausgearbeitet [DLR IfEV08], der die Zusammenarbeit zwischen dem IfEV und dem gesamten DLR regelt.

Verbesserung der Empfangswahrscheinlichkeit durch Aufteilung des Empfangsbereiches

Auf dem Turm befindet sich die dreh- und schwenkbare Richtantenne eines Flugzeug-Verfolgungs-Datalink [Jessen]. Auf der oberen Plattform kann somit keine Antenne aufgebaut werden. Daher konnte auf dem Turm keine Omnidirektionalantenne aufgestellt werden, sondern es mussten zwei Sektorantennen an den Seiten des Turmes angebracht werden (Bild 36).



Bild 36: Data-Link-Turm des DLR

Bei der zu erwartenden Reichweite erschien es ohnehin zweckmäßig, den Empfangsbereich (die norddeutsche Tiefebene) in zwei Sektoren aufzuteilen und somit die Funkfeldlast der einzelnen Empfänger zu halbieren. Das heißt, die Anzahl von Nachrichten pro Empfänger wird halbiert und somit die Häufigkeit von Nachrichtenüberlagerungen verringert.

Auf der zweiten Plattform von oben befindet sich ein geschlossener Raum, der die Sende- und Empfangsanlagen des Datalinks aufnimmt. Darin konnten die SPUs untergebracht werden. Auf dieser Plattform konnten die Antennen allerdings nicht angebracht werden, da die Umwandlung nicht

verletzt werden durfte. Die nächst tiefere Plattform ist begehbar, dort konnten die Antennen installiert werden. Die Antennen sind in 22 m Höhe am Stahlurm (108 m über Normal Null) befestigt und damit 12 m höher als die Antenne auf dem Dach des IfEV, so dass die umgebenden Wälder überragt werden.

Es kommen L-Band Sektor Antennen der Fa. Elta Electronic Industries (eine Tochtergesellschaft der IAI Electronic Group) mit Radomen zum Einsatz, welche aus Altbeständen der Fa. Thales stammen. Diese Antennen besitzen eine horizontal halbkreisförmige Charakteristik mit 200° Öffnungswinkel und werden somit nicht von Reflexionen am Turm beeinträchtigt. Außerdem besitzen die Antennen eine Vertikalcharakteristik, die bei Elevation 0° eine Nullstelle haben und bis 35° Elevation einen Gewinn von +15 dB liefern. Am Institut wurden solche Antennen schon in den Projekten des Sonderforschungsbereiches 58 und 212 [Form90] der Deutschen Forschungsgemeinschaft eingesetzt.

Um die Antennen windfest anbringen zu können, wurde ein mechanisches Befestigungssystem von Prof. Form am IfEV selbst konstruiert und angefertigt.

Aufbau der Doppelstation

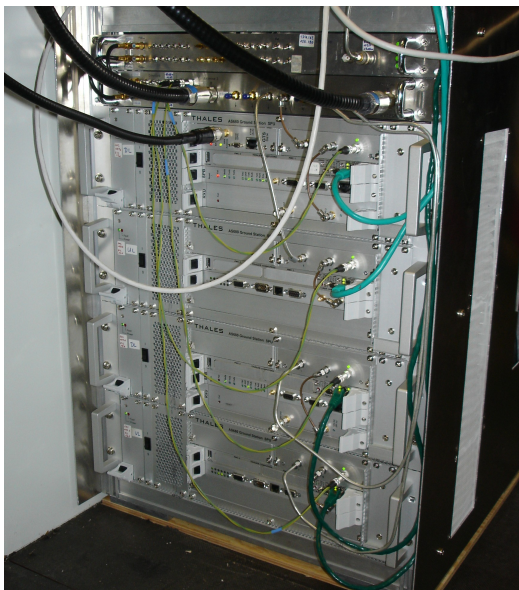


Bild 37: Doppelstation auf dem DLR-Turm mit Testsignalgenerator, Frontend und SPUs

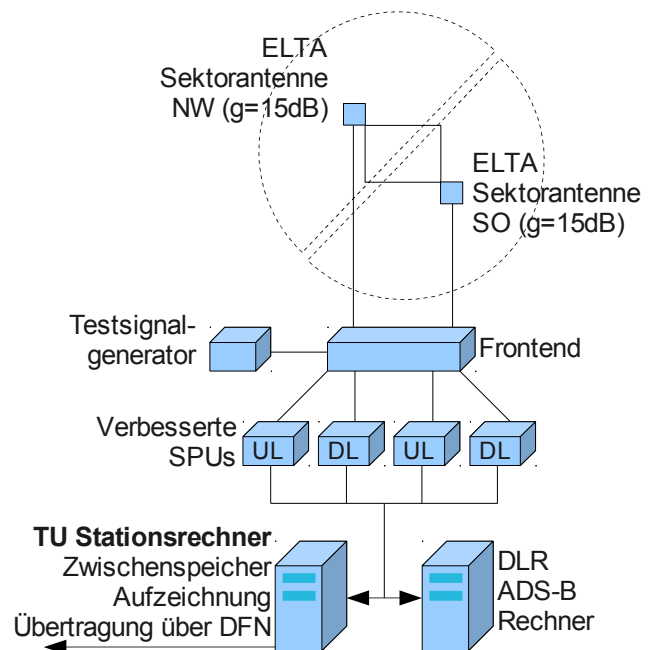


Bild 38: Schematischer Aufbau der Doppelstation im DLR Braunschweig

Diese Station besteht aus zwei SPU-Paaren und bildet damit eine Doppelstation. Eine Sektor-Antenne ist nach Nordwesten gerichtet, eine nach Südosten. An jede Antenne ist ein 1030MHz/1090MHz-SPU-Paar angeschlossen. Zwischen Antennen und SPUs ist ein vom IfEV entwickeltes Frontend eingefügt, dass die nötigen Anpassungen der Empfänger an die ELTA Antennen vornimmt und Testsignale (siehe Abschnitt 4.3.1) in die Antennenleitungen einspeist.

Die vier SPUs übermitteln die Empfangsdaten über Ethernet an den Stationsrechner, der sich im Keller des DLR-Institutes befindet (Bild 38). Es ist auch vorgesehen, einen DLR-Rechner zur Aufzeichnung der ADS-B-Daten dort aufzustellen. Der (oder die) Stationsrechner sind über das

sogenannte Fremdfirmen-LAN des DLR [DLR F-LAN] an das Internet angebunden. Die Station befindet sich somit nicht im DLR-internen Netz und ist vom IfEV aus über das Internet administrierbar, ohne eine Sicherheitslücke für das DLR-Netz darzustellen.

Reichweite der neuen Doppelstation auf dem DLR-Turm

Um festzustellen, ob der Empfang sich durch die Doppelstation verbessert hat, wurden die Reichweiten der Stationen verglichen. Dies geschah durch Vergleich der empfangenen ADS-B-Spuren. Bild 39 zeigt einen mit der Antenne auf dem Institutsdach aufgezeichneten ca. zehnminütigen Mitschnitt von ADS-B-Spuren vom 03. Juni 2008. Der Empfangsbereich der Anlage auf dem TU-Institutsdach ist umrandet. Bild 39 vermittelt einen Eindruck in welchen Entfernungen noch zusammenhängende Spuren beobachtet werden, wo nur noch vereinzelte Positionsmeldungen empfangen werden und in welchen Entfernungen keine Nachrichten mehr empfangen werden können. Die Beschriftungen der Spuren spielen hier keine Rolle.

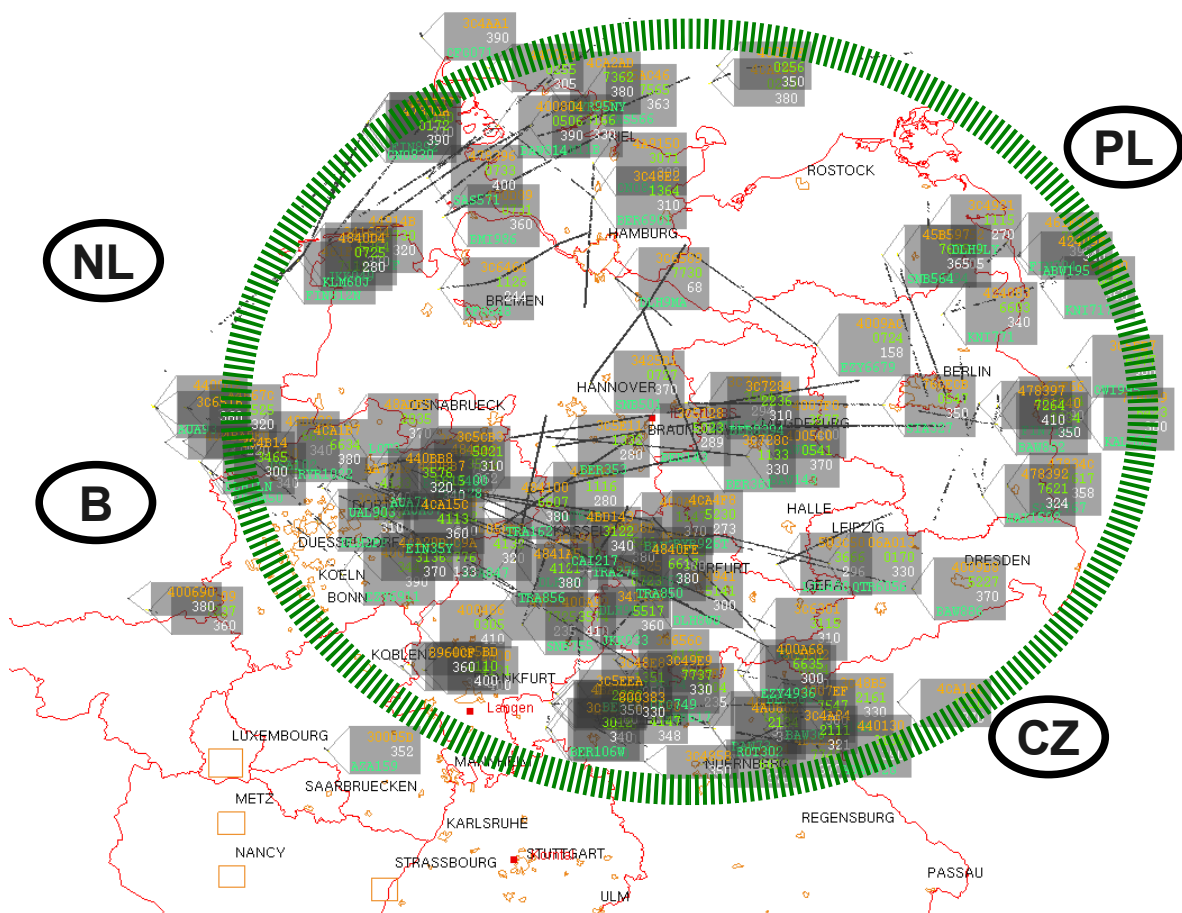


Bild 39: Reichweite der Station auf dem Institutsdach des IfEV

Bild 40 zeigt die ADS-B-Spuren, die mit der Anlage auf dem DLR-Turm im gleichen Zeitraum empfangen wurden. Der Empfangsbereich der Institutsdach-Anlage ist wieder markiert. Es ist zu sehen, dass sich der Empfangsbereich vergrößert hat und insbesondere im Südwesten deutlich mehr Flugzeuge zu sehen sind. Der Wald im Südwesten schattet die Station am Turm im Gegensatz zur Station auf dem Institutsdach nicht mehr ab.

Gleiches zeigt sich auch für einen Anflug auf den Flughafen Hannover. Vom Institutsdach lassen sich Anflüge bis 1000 ft über NN (in Screenshot des Thales ADS-B-Display im Bild 41 als FL 10 dargestellt, dritte Zeile im Label) beobachten. Vom DLR Turm lassen sich Anflüge weit tiefer bis 400 ft über NN beobachten (dargestellt als FL 4 in Bild 42). .

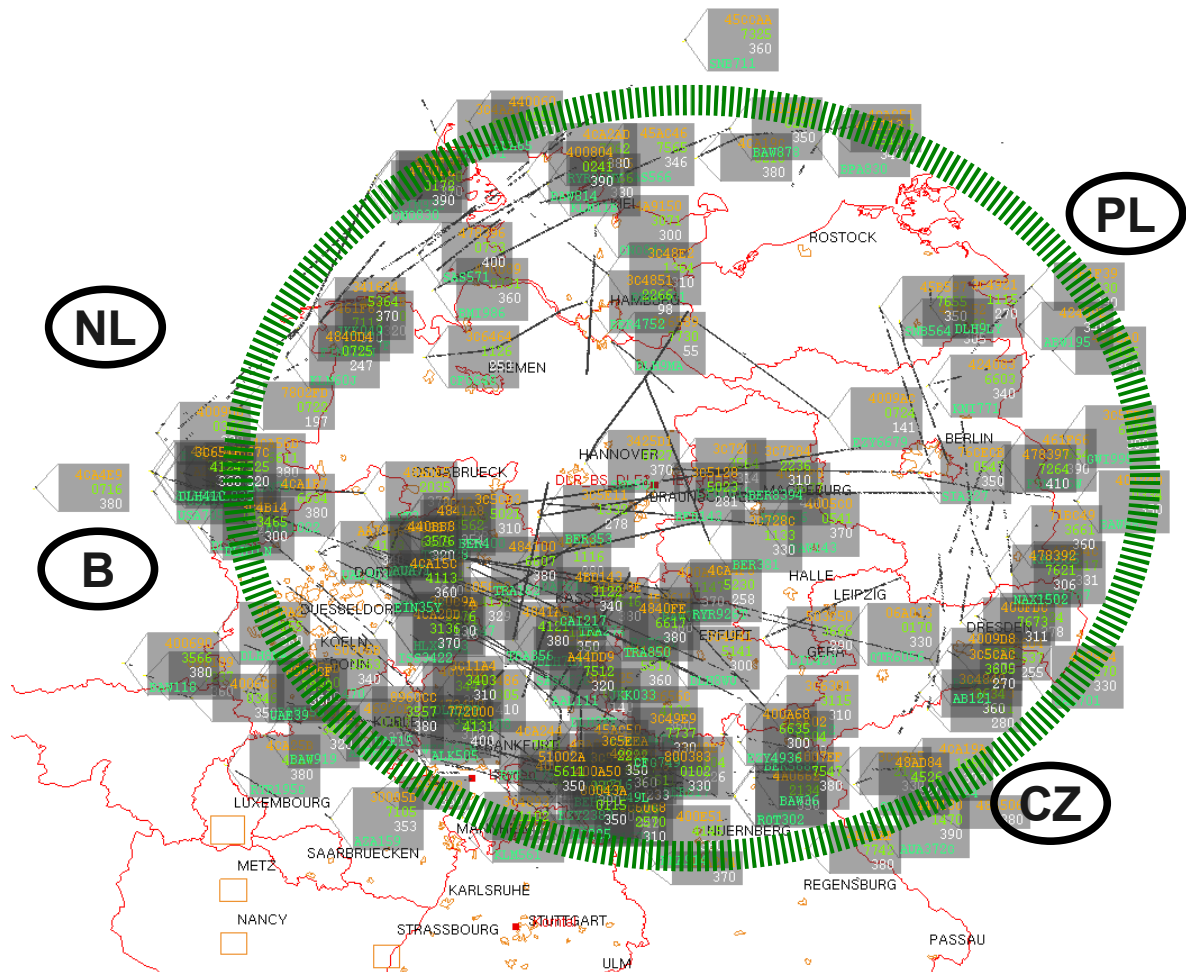


Bild 40: Reichweite der Station auf dem DLR-Turm

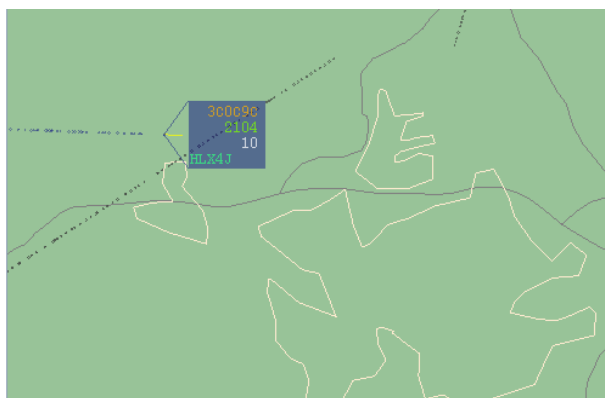


Bild 41: Anflug auf Hannover beobachtet vom TU-Dach

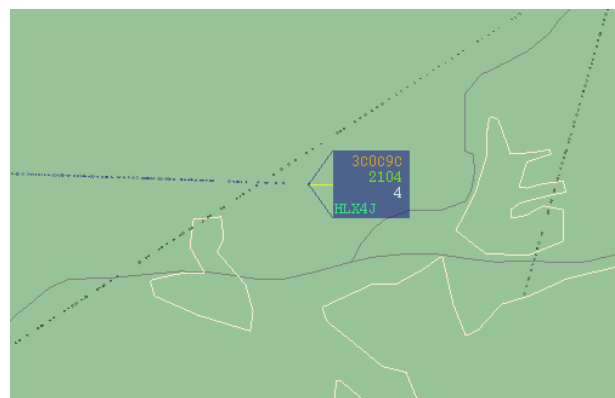


Bild 42: Anflug auf Hannover beobachtet vom DLR-Turm

8.2.3 Standort 2: FRAPORT

Da Frankfurt der verkehrsreichste Flughafen in Deutschland ist, ist eine Station dort für das ACAS-Monitoring von zentraler Bedeutung. Die Fraport AG konnte gewonnen werden, eine Station aufzunehmen. Als Standort wurde Gebäude 101 ausgewählt, da es die umliegenden Gebäude überragt und dort auch keine störenden Radaranlagen oder Mobilfunkmasten in unmittelbarer Nähe sind. In Bild 43 ist Gebäude 101 markiert, die Antenne (Bild 44) befindet sich bei $N50^{\circ} 03.090'$ $E8^{\circ} 35.714'$. Für die SPU und den Stationsrechner (Bild 45) wurde eine Kammer unter der Antenne gemietet. Der beobachtbare Bereich ist in Bild 46 gezeigt.

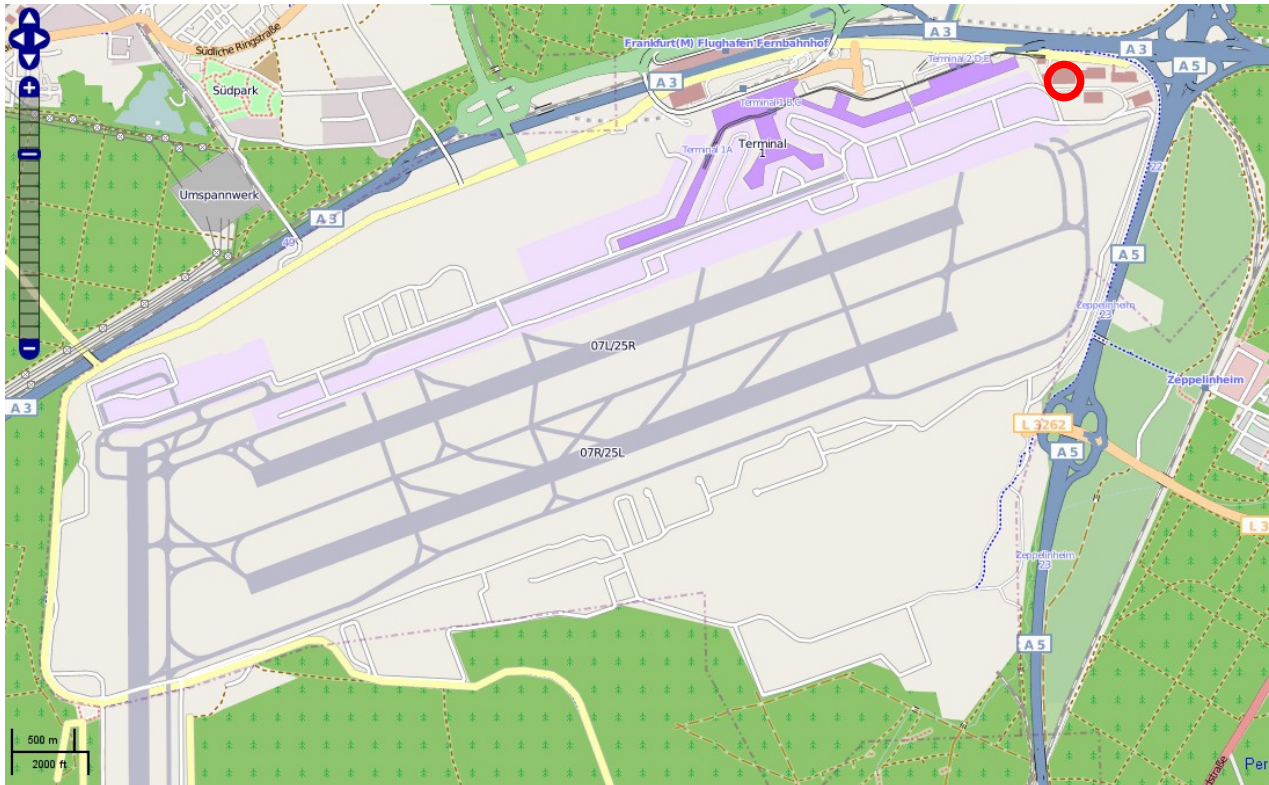


Bild 43: Standort der Station am Flughafen Frankfurt [OSM]

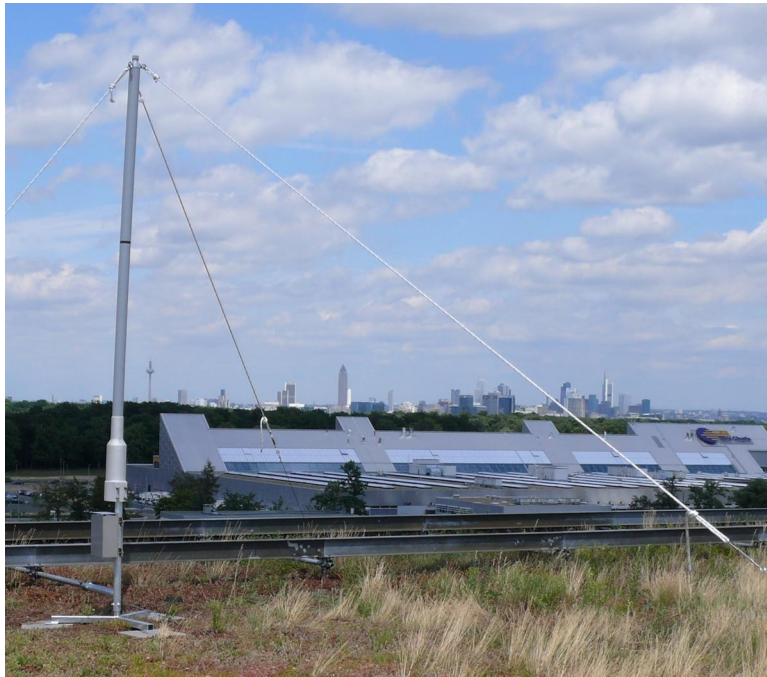


Bild 44: Antenne in Frankfurt auf Gebäude 101 (begrüntes Dach), im Hintergrund Skyline von Frankfurt



Bild 45: Uplink- und Downlink SPU, Stationsrechner und USV in Frankfurt, gleicher Aufbau in Oberpfaffenhofen und Korntal

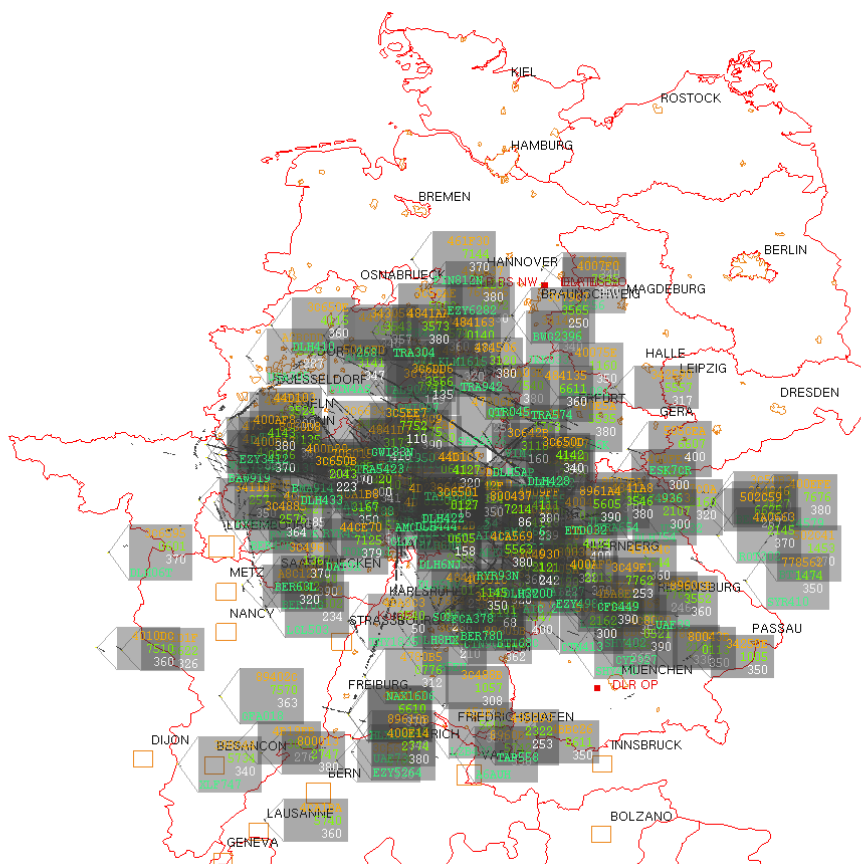


Bild 46: Reichweite der Station in Frankfurt

8.2.4 Standort 3: DLR-Oberpfaffenhofen

Um den süddeutschen Luftraum und den Verkehr um München beobachten zu können, wurde ein Standort in Süddeutschland gesucht. Das DLR-Oberpfaffenhofen erklärte sich bereit, eine Station aufzunehmen. Die Antenne befindet sich bei $N48^{\circ} 04.896' E11^{\circ} 16.556'$ (Bild 47).

Leider steht die Station noch nicht am endgültigen Ort. Zum einen wird der Empfang der Antenne durch einen Wald im Nordwesten behindert, zum anderen wird befürchtet, dass die Antenne ein Experiment des DLR stören könnte. Die Station wird deshalb verlegt werden. Der Empfangsbereich der Station ist in Bild 48 dargestellt.



Bild 47: Empfangsanlage am Flughafen Oberpfaffenhofen

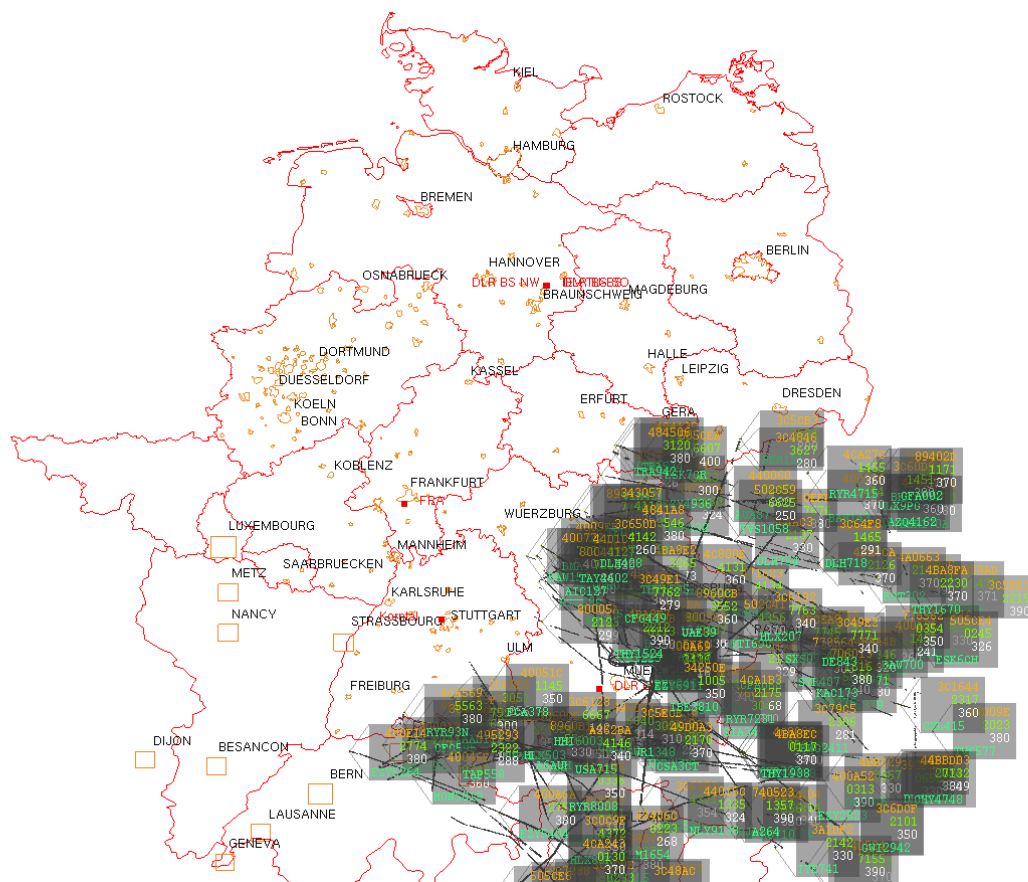


Bild 48: Reichweite der Station in Oberpfaffenhofen

8.2.5 Standort 4: Thales, Korntal bei Stuttgart

Es bot sich an, eine Station bei der Herstellerfirma Thales, die auch Partner im SATT-Projekt war, aufzustellen. Zwar hat Thales Stuttgart seinen Firmensitz (N48° 49.537' E09° 06.680', Bild 49) nicht im Tal bei Stuttgart, aber auch nicht auf einer Bergkuppe. Der Standort ist somit nicht perfekt. Dennoch ist die Station ein wichtiger Teil des ACAS-Monitor-Netzes wie Bild 50 zeigt.



Bild 49: Großraum Stuttgart mit Standort der Empfangsanlage und Flughafen [OSM]

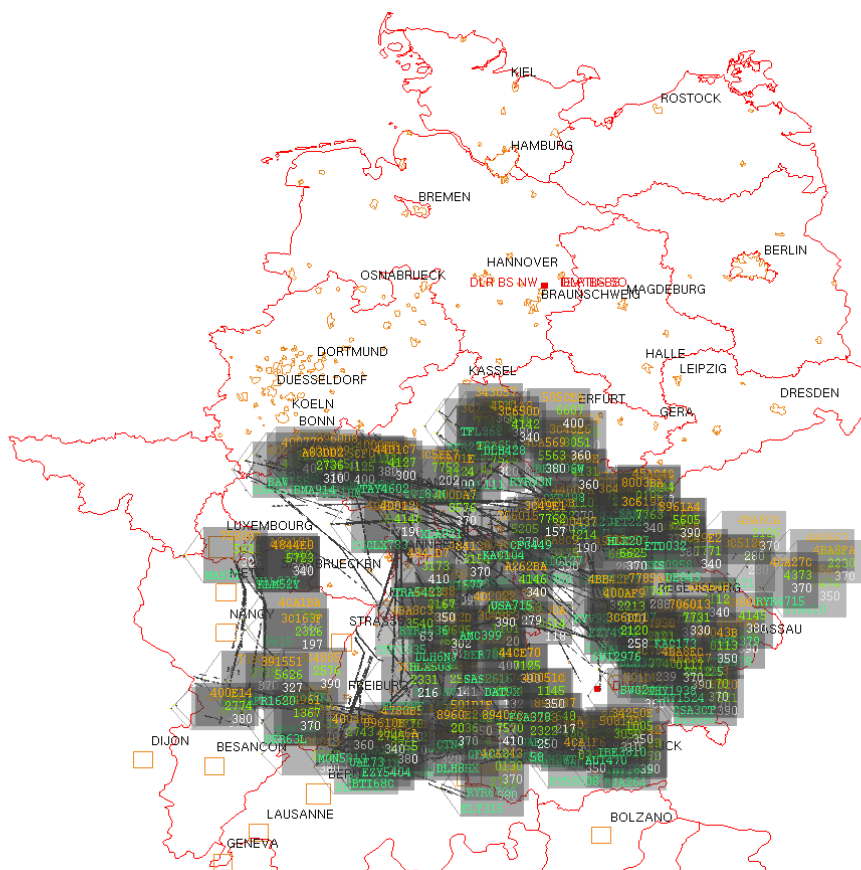


Bild 50: Reichweite der Kornthaler Station

8.3 Reichweite des Netzes

Werden Asterix-Positionsdaten der fünf Stationen kombiniert, so ergibt sich Bild 51. Darin ist zu sehen, dass zumindest im oberen Luftraum ganz Deutschland und Teile der Nachbarländer bedeckt werden.

Um realzeitlich ein Verkehrslagebild von ganz Deutschland zu erhalten, werden die Positionsdaten bereits in den Stationen aus den Mode S-Daten extrahiert und sofort, ohne Pufferung per UDP nach Braunschweig gesendet. Ist das Netz überlastet oder zusammengebrochen, werden die UDP-Pakete verworfen und nicht noch einmal übertragen. Die kompletten Mode S-Aufzeichnungen, die auch die Positionsdaten enthalten, werden parallel dazu datiert gepackt und bei Bedarf für Wochen in den Stationen gepuffert. Die für die reinen Positionsmeldungen nötige Datenrate ist ca. 1 kByte/s pro Station. Die Rate der kompletten Mode S-Empfangs-Daten ist ungleich größer. Wie viele Daten verarbeitet werden mussten und wie das geschieht, wird im nächsten Kapitel dargestellt.

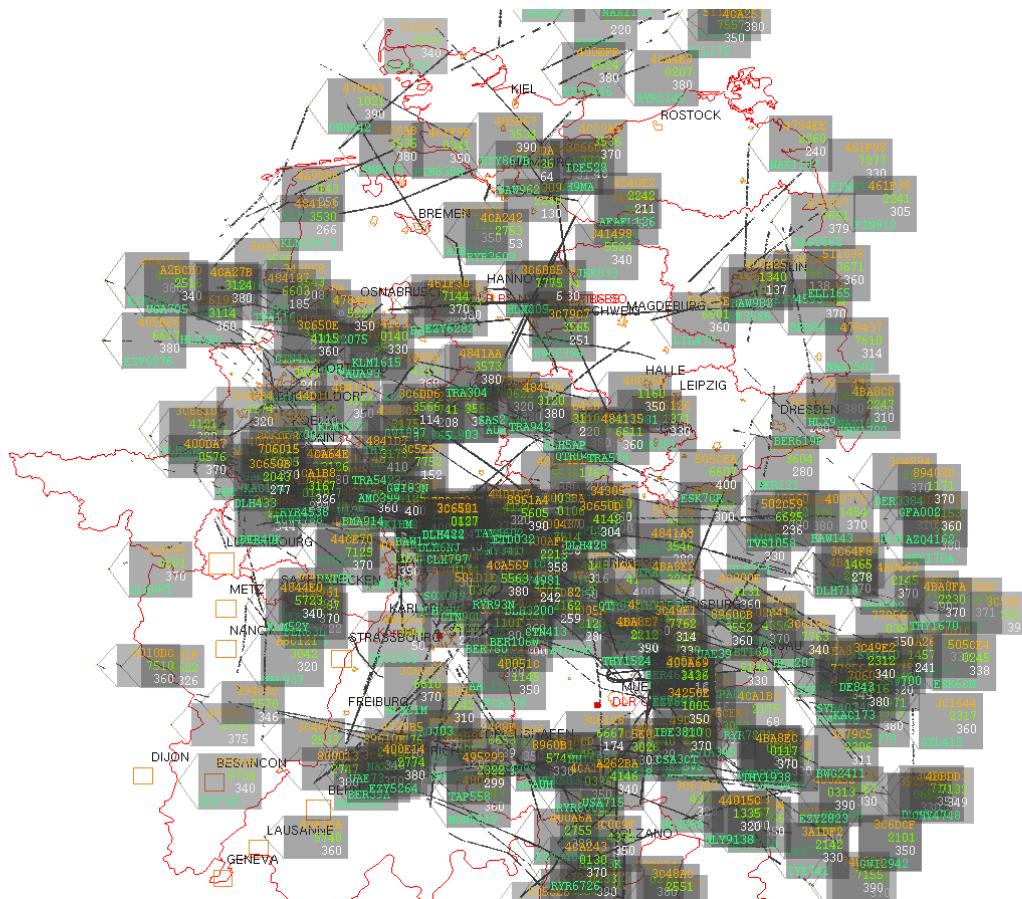


Bild 51: Reichweite des ACAS-Monitor-Netzes

8.4 Datenübertragung von SPUs zur Zentrale

Die empfangenen Mode S-Daten müssen komplett zur Zentrale in Braunschweig übertragen werden. Dabei war die Frage zu beantworten, in welchem Format und auf welchem Übertragungsweg die Daten nach Braunschweig gelangen sollten.

- Wichtig ist, dass bei der Übertragung **keine Daten verloren** gehen. Aus den empfangenen Daten sollte ein Maximum an Informationen extrahiert werden können.

- Die **Empfangszeitpunkte** der Mode S-Nachrichten müssen mit gespeichert werden, damit Totzeiten bei der Übertragung und der Analyse keine Rolle spielen. Außerdem sollen während der Entwicklung des ACAS-Monitors archivierte Daten später mit überarbeiteter ACAS-Monitor-Software erneut analysiert werden können.
- Die Standorte sind zu weit von Braunschweig entfernt, um bei Bedarf „mal eben“ vor Ort Probleme lösen zu können. Die Stationen müssen also **fernwartbar** sein.
- Die Stationen, und vor allem ihre Netzwerkanbindungen, müssen so **robust** sein, dass sie sich auch bei Teilausfällen noch fernwarten lassen. Und sie müssen nach Störungen möglichst selbstständig wieder in den normalen Betriebszustand gehen.
- Dass die Daten **schnell** nach Braunschweig übertragen werden, ist wünschenswert, aber nicht zwingend. Echtzeitanwendungen, wie die Anzeige von RA-Informationen und ADS-B-Echtzeit-Übertragung sind angedacht und können nachträglich eingebaut werden.

8.4.1 Vorbereitende Testaufzeichnung

Um die Datenraten, die übertragen und verarbeitet werden sollen, abzuschätzen, konnten die DFS und Thales gewonnen werden, mit ihren eigenen Empfangsanlagen eine gemeinsame Aufzeichnung vom 12. Juli 2007 15:30 Uhr bis 13. Juli 2007 24:00 Uhr durchzuführen. Die Daten wurden per DVD nach Braunschweig geschickt und dort mit den Braunschweiger Aufzeichnungen der experimentellen Anlage ausgewertet. Die Auswertungen wurden für jede Station einzeln und für den gemeinsamen zusammengefassten Datenstrom (Spalte „Gesamt“ in Tabelle 7) durchgeführt.

	Braunschweig	Langen	Korntal	Gesamt
Uplink-Nachrichten [1/s]	500	1250	1000	2750
Korrekte Uplink Nachrichten [1/s]	450	1100	600	2250
Downlink-Nachrichten [1/s]	1400	800	800	3000
Korrekte Downlink Nachrichten [1/s]	600	700	700	2000
Maximale Anzahl von Mode S-ausgerüsteten Flugzeugen, gleichzeitig im Empfangsbereich	250	400	250	550
Maximale Anzahl ACAS-Flugzeuge gleichzeitig im Empfangsbereich	240	370	220	470
Maximale Anzahl ADS-B-Flugzeuge gleichzeitig im Empfangsbereich	170	250	150	310
Anzahl der ACAS-Events	2	0	0	2
Anzahl der ACAS-Events zwischen 11.07. 12:30 bis 16.07. 12:00 (In diesem Zeitraum waren nicht alle Stationen gleichzeitig in Betrieb)	14	2	4	16
Datenrate während der Hauptverkehrszeiten Up- u. Downlink, ungepackt aus obigen Werten berechnet Thales Anlage: 88 Byte/Nachricht TU Braunschweig Anlage: 108 bzw. 122 Byte/Nachricht	210 kByte/s	180 kByte/s	155 kByte/s	545 kByte/s
Datenrate (Up- u. Downlink) ungepackt, aus Dateigröße bestimmt über den jeweiligen Aufzeichnungszeitraum gemittelt	nicht bestimmt	120 kByte/s	120 kByte/s	nicht bestimmt
Datenrate (Up- u. Downlink) gepackt (bz2), aus Dateigröße bestimmt über den jeweiligen Aufzeichnungszeitraum gemittelt	24 kByte/s	13 kByte/s	9 kByte/s	46 kByte/s

Tabelle 7: Gemeinsame Test-Aufzeichnung in Braunschweig, Korntal und Langen

Die Ergebnisse der Testaufzeichnung sind in Tabelle 7 zusammengefasst. Es zeigt sich, dass sich die Empfangsbereiche der Stationen überlappen und ein Flugzeug von mehr als einer Station gleichzeitig beobachtet werden kann und auch mehrere Stationen zu einem ACAS-Event beitragen können. Daher ist die Spalte „Gesamt“ nicht einfach die Summe der anderen Spalten.

Außerdem konnte die Datenrate abgeschätzt werden. In den Hauptverkehrszeiten wurden in Langen 180 kByte/s Mode S-Daten aufgezeichnet. Die Daten waren im ungepackten Textformat von Thales codiert. Dieses Textformat braucht 88 Byte pro Nachricht. Es ist damit sehr ineffizient und stellt somit eine obere Schranke für die zu erwartende Übertragungsrate dar. Bei den gepackten Daten ergab sich in Langen, über den ganzen Tag gemittelt, eine Datenrate von 13 kByte/s. Bei einem Sicherheitsfaktor von zwei müssten also wenigstens ca. 30 kByte/s übertragen werden können, um über den Tag gemittelt alle Aufzeichnungen zur Zentrale in Braunschweig zu übertragen. Für Realzeitanwendungen sind dagegen ca. 300 kByte/s nötig, um auch zu den Hauptverkehrszeiten die Daten schnell zur Zentrale zu bringen.

8.4.2 Übertragungsverfahren: LAN der Partner oder ADSL

Die Stationen sollten also möglichst verlässlich und preiswert Daten nach Braunschweig übermitteln können. Für das geplante zeitversetzte (Architektur 2, Abschnitt 8.1.2) ACAS-Monitoring sollten mindestens ca. 30 kByte/s Daten übertragen werden können.

Optimal ist natürlich eine direkte Anbindung der Stationen über die Firmen-LANs der Partner. War dies nicht möglich, wurden ADSL-Anschlüsse beantragt. ADSL-Anschlüsse sind aufgrund des massenhaften Privatkundengeschäfts ausgereift und preisgünstig. In Frankfurt wurde ein ADSL-Anschluss mit 500 kByte/s (4000 kbit/s) Downstream und 95 kByte/s Upstream geschaltet, in Korntal ein Anschluss mit 430 kByte/s (~3500 kbit/s) Downstream und 56 kByte/s Upstream. Die ADSL-Anschlüsse werden „falsch-herum“ betrieben, der kleinere Upstream wird genutzt, um die Empfangsdaten zu übertragen. In diesem Preissegment bieten ADSL-Provider nur solche Anschlüsse an. Alternativ sind auch WLAN-Brücken oder UMTS-Anbindungen möglich.

8.4.3 Stationsrechner als Zwischenspeicher

An jedem Standort steht eine Thales-Uplink-SPU und eine Thales-Downlink-SPU. Die SPU's geben die Mode S-Daten im Thales Binärformat (siehe 4.2.1) über UDP aus. Damit kümmert sich die SPU nicht darum, ob die Daten auch beim Empfänger ankommen. Der Empfänger kann nicht überprüfen, ob alle Daten empfangen wurden. Außerdem werden die Daten auf der SPU nicht gepuffert. Da der zentrale Server in Braunschweig und auch die Internetverbindung ausfallen oder überlastet sein können, sollten die Daten am Standort gepuffert werden und per TCP (mit Datenflusskontrolle) übertragen werden. Authentifizierung und Verschlüsselung wären ebenfalls wünschenswert. Die Embedded Linux Systeme in den SPU's ließen sich (mit hohem Aufwand) so erweitern, dass sie die Mode S-Daten per TCP verschlüsselt senden. Um große Datenmengen zu puffern, fehlt ihnen allerdings der Speicher.

Aus diesen Gründen wurde an jedem Standort ein zusätzlicher Computer aufgestellt. Diese Stationsrechner sind vollwertige Linux-Systeme (Thales verwendet die Distribution Fedora Core 6 [Fedora]) mit vollständigen Toolchains³². Damit lässt sich auf ihnen leicht die Software zur Daten-

³² Toolchain: Kette von Entwicklungswerkzeugen zum Erstellen, Modifizieren und Installieren von Programmen.

pufferung und Wartung der SPUs installieren. Zur Wartung der SPUs kann sich der Administrator über das Internet (per SSH [SSH]) oder vor Ort an den Stationsrechnern anmelden, auf denen die Wartungssoftware (Thales Technical System Control) für die SPUs installiert ist. Die Mode S-Daten werden von den SPUs an die Stationsrechner übermittelt, wo sie von der in dieser Arbeit entwickelten Software Thudp2binfile (Abschnitt 4.2.4) aufgezeichnet werden. Um SPUs und Stationsrechner gemeinsam an einen ADSL-Anschluss oder an eine IP-Adresse im Netz der Partner anzubinden, wird ein NAT³³-Router eingesetzt.

Die Stationen an den fünf Standorten sind alle ähnlich aufgebaut (Bild 52). An einer omnidirektionalen Antenne sind die Thales SPUs für Uplink und Downlink angeschlossen. In Braunschweig sind 4 SPUs mit 2 Sektorantennen an den Stationsrechner angeschlossen.

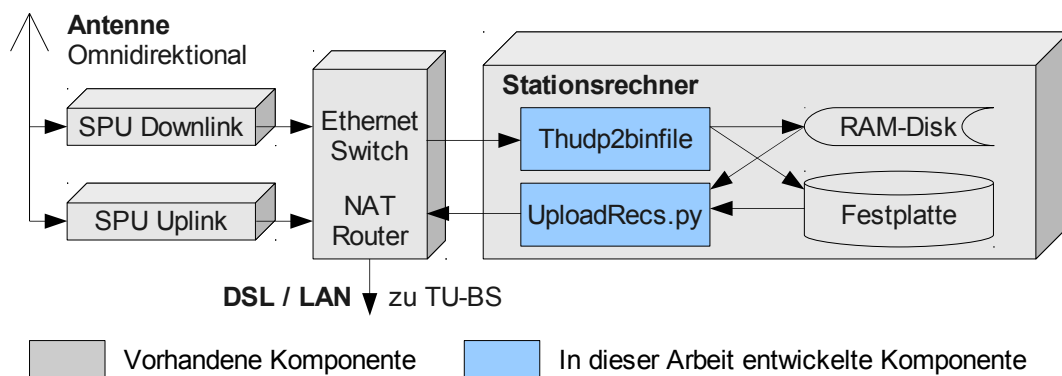


Bild 52: Schematischer Aufbau einer Station

8.4.4 Ablauf in den Stationsrechnern

Die Stationsrechner sollen die empfangenen Daten von den SPUs aufzeichnen, mit einem Datum versehen, komprimieren, puffern und wenn möglich an die Zentrale übertragen.

Um Mode S-Daten von den SPUs aufzuzeichnen, liefert Thales einige Programme mit. Diese müssen allerdings über eine grafische Benutzerschnittstelle bedient werden. Das für die Automatisierung wichtige Kommandozeileninterface ist zwar vorhanden, funktioniert aber nicht. Um die Thales Software zu ersetzen und um weitere eigene Anforderungen zu realisieren, musste daher Thudp2binfile komplett neu entwickelt werden. Thudp2binfile nimmt die binären Mode S-Daten von einer SPU entgegen, decodiert den Datenstrom, fügt das Empfangsdatum hinzu und packt die Daten in handhabbare Dateien. Damit ist aus dem Empfangsdatenstrom eine Folge von Dateien geworden. Die Dateien lassen sich einfach über Internet kopieren, zur Analyse entpacken und gemeinsam archivieren.

Die Dateien werden von Thudp2binfile beim Schreiben mit Bzip2 komprimiert, mit Anfangs- und Endzeitpunkt benannt und abgelegt. Sind in eine Datei 10 MByte Daten geschrieben, so wird eine neue Datei begonnen – damit behalten die Dateien eine handhabbare Größe. Diese fertigen Dateien verschiebt dann das Skript UploadRecs.py mittels SCP³⁴ nach Braunschweig.

Um die Festplatte im Stationsrechner nicht durch dauerndes Schreiben und Löschen vorzeitig zu verschleifen, werden die Aufzeichnungen von Thudp2binfile in einer Ramdisk³⁵ abgelegt und von

33 NAT (Network Address Translation): Technik, um mehrere Netzwerkgeräte über eine IP-Adresse ans Internet anzubinden, wird oft verwendet, wenn vom Zugangsprovider nur eine IP-Adresse zur Verfügung gestellt wird.

34 SCP ist ein Datei-Transferprotokoll das auf SSH aufbaut.

35 Ramdisk ist ein virtueller, nicht persistenter Datenträger im Arbeitsspeicher eines Computers, der keiner Abnutzung unterworfen

dort von UploadRecs.py entnommen. Ist diese Ramdisk voll, z. B. weil die Daten wegen eines Netzwerkproblems nicht übertragen werden können, speichert Thudp2binfile automatisch auf die lokale Festplatte. Wenn das Übertragungsproblem behoben ist, entnimmt UploadRecs.py die Daten auch von dort und wird bald sowohl die Festplatte als auch die Ramdisk geleert haben. So können Netzerkausfälle von mehreren Wochen überbrückt werden. Und nur so ist der Betrieb des ACAS-Monitor-Netzes der TU Braunschweig durch eine Person möglich.

8.5 Sicherheitskonzept des ACAS-Monitor-Netzes

Da das ACAS-Monitor-Netz mit dem Internet verbunden ist, müssen Stationen und Zentrale gegen Eingriffe Dritter über das Internet geschützt werden. Dazu wurde folgendes Sicherheitskonzept erarbeitet.

Das Sicherheitskonzept des ACAS-Monitor-Netzes beruht auf SSH [SSH] – und dabei vor allem darauf, dass die beteiligten Computer (Server, Entwicklungsrechner und Stationsrechner) **ausschließlich** per SSH zugänglich sind. SSH (implementiert durch das OpenSource-Produkt OpenSSH [OpenSSH]) wird dabei zur Verschlüsselung, zur Authentisierung und zum Tunneln³⁶ aller Kommunikationen benutzt. Die Firewalls und NAT-Router in den Stationen und der Zentrale lassen lediglich SSH-Zugriffe zu. So gibt es nur einen von außen zugänglichen und damit verwundbaren Dienst. Nur dieser Dienst muss überwacht und die Sicherheitsmeldungen in den einschlägigen Medien beobachtet werden.

Die typischen Angriffsvektoren auf SSH sind zum einen Wörterbuchattacken, bei denen der Angreifer viele (bis zu mehrere tausend) Nutzernamen-Passwort-Kombinationen ausprobiert, und zum anderen zukünftig entdeckte Sicherheitslücken im SSH-Dienst, bei denen Programmierfehler im SSH das Eindringen in das System ermöglichen.

Überwachung von SSH und Sicherung gegen Wörterbuchattacken

Die meisten Angreifer versuchen sich per SSH als Systemadministrator „root“ einzuloggen, weil dieses Nutzerkonto auf (fast) allen Linux-Systemen vorhanden ist. Daher ist auf allen Computern im ACAS-Monitor-Netz das Einloggen als root per SSH verboten. Damit ist jeder Versuch, das root-Passwort zu raten, unterbunden.

Um Wörterbuchattacken zu erkennen und um den ganzen SSH-Zugang zu überwachen, wurde auf den Computern der Dienst Denyhosts installiert. Denyhosts analysiert die SSH-Zugriffsprotokolle zeitnah und blockiert IP-Adressen, von denen wiederholt erfolglose Loginversuche unternommen wurden. Täglich werden im Mittel zwei Wörterbuchattacken erkannt. Des Weiteren sind (fast³⁷) alle Computer so konfiguriert, dass SSH das Einloggen per Nutzernamen-Passwort-Kombination verbietet. Die Nutzer müssen sich dann mit ihrem privaten Schlüssel authentisieren, was Wörterbuchattacken praktisch unmöglich macht.

ist wie eine Festplatte.

36 Tunneln: Übertragungsart von Netzwerkpaketen, indem diese nicht über die darunterliegende Netzwerktransportschicht übertragen werden, sondern in Pakete eines anderen Protokolls verpackt werden. Damit kann z. B. nachträglich eine Verschlüsselung zu einem nicht verschlüsselnden Netzwerkprotokoll hinzugefügt werden.

37 Noch nicht komplett umgesetzt, da noch nicht alle Stationen den Zugriff per Passwort verbieten.

Sicherheitsupdates für SSH

Um neu gefundene Sicherheitslücken im SSH zu bemerken, werden die einschlägigen Medien (mindestens Heise-Security [Heise.de]) verfolgt.

So wurde im Herbst 2008 eine Sicherheitslücke im verwendeten OpenSSH-Paket der Debian [Debian] Linux-Distribution entdeckt [Debian SA1571-1]. Diese Linux-Distribution wird auf dem zentralen Server und den Entwickler PCs verwendet. Die fehlerhafte Version des OpenSSH-Paketes erzeugte leicht zu erratende Schlüssel, d. h. das fehlerhafte Paket konnte nur 32768 verschiedene Schlüssel erzeugen, was einem Angreifer das Erraten eines Schlüssels zu leicht macht. Der Host-Key des zentralen Servers war betroffen und wurde umgehend ausgetauscht. Mit dem Host-Key identifiziert sich der Server gegenüber den Entwicklern und Administratoren, und gegenüber den Stationen. Einem Angreifer wäre es unter Umständen möglich gewesen, die Entwickler, Administratoren oder Empfangsdaten auf ein anderes System umzuleiten. Es gab aber keine Anzeichen für solch einen erfolgten Angriff.

Minimierung von Zugriffsrechten

Für die verschiedenen Aufgaben im ACAS-Monitor-Netz gibt es verschiedene Nutzerkonten. Alle Nutzerkonten im ACAS-Monitor-Netz haben minimale Zugriffsrechte, so fällt es einem etwaigen Angreifer schwerer, von einem Nutzerkonto aus das komplette Netz zu unterwandern.

Die Stationsrechner müssen sich automatisch auf dem Server einloggen, um Mode S-Aufzeichnungen abzuliefern. Dabei hat jede Station ihr eigenes Nutzerkonto mit eigenem Schlüssel und darf nur ihr eigenes Eingangsverzeichnis schreiben und lesen. Ein Prozess auf dem Server entnimmt die eingegangenen Dateien, entzieht den Stations-Nutzerkonten die Rechte und sortiert die Dateien in Verzeichnisse zur Auswertung und Archivierung. Die Rechte an den Dateien werden dem Nutzerkonto zur automatischen Auswertung übertragen. So wird verhindert, dass ein erfolgreicher Einbruch in eine Station im Feld das ganze Netz kompromittieren kann.

Die Entwicklung der Software findet auf eigenen Entwicklungsrechnern mit eigenen Nutzerkonten statt. Softwareänderungen lassen sich durch Verwendung des Versionsverwaltungssystems SVN [SVN] auf Nutzerkonten zurückführen.

8.6 Automatisierte Auswertung in der Zentrale in Braunschweig

Die Stationen laden ihre Aufzeichnungsdateien in Eingangsverzeichnisse auf den zentralen Server in Braunschweig. Von dort entnimmt sie ein Prozess (`take_files_form_inboxes.py`) und sortiert sie in Verzeichnisse für jeden Tag. Ein solches Verzeichnis umfasst alle Dateien von 23:55 Uhr des Vortags bis 00:05 Uhr des Folgetages. Wird eine Datei für zwei Tage benötigt, wird sie nur im Verzeichnis des ersten Tages abgelegt und im zweiten darauf verwiesen, um Speicherplatz zu sparen. Das Verzeichnis eines gewöhnlichen Tages, willkürlich wurde der 20. Mail 2009 gewählt, ist in Listing 6 dargestellt.

Es werden Unterverzeichnisse für jede Station angelegt. Die Namen der Aufzeichnungsdateien enthalten die Kennung der Station, die Kennung des Kanals (Uplink, Downlink, außer in Braunschweig, wo die Kanäle bereits in der Station vereinigt werden) und Start- und Endzeitpunkte der Dateien. In jede Datei werden 10 MByte unkomprimierte Daten geschrieben, was komprimiert ca.

4 MByte ergibt. Die komplette Aufzeichnung des 20. Mai 2009 umfasst 1001 Dateien oder ca. 4,1 GByte. Dieser Speicherbedarf ist auch die Regel für jeden anderen (Werk-)Tag ohne besondere Vorkommnisse wie z. B. Stationsausfälle.

```
20090520
|-- dlrbs
|   |-- Modes_dlrbs_20090519T234933_to_20090520T000353.UTC.thxbin.bz2
|   |-- Modes_dlrbs_20090520T000353_to_20090520T001730.UTC.thxbin.bz2
|   |-- ... 391 weitere Dateien ...
|   `-- Modes_dlrbs_20090520T234731_to_20090521T000247.UTC.thxbin.bz2
|-- dlrop
|   |-- Modes_dlrop_dl_20090519T175008_to_20090520T100636.UTC.thxbin.bz2
|   |-- Modes_dlrop_dl_20090520T100636_to_20090520T101226.UTC.thxbin.bz2
|   |-- ... 129 weitere Dateien ...
|   |-- Modes_dlrop_dl_20090520T232553_to_20090521T000746.UTC.thxbin.bz2
|   |-- Modes_dlrop_ul_20090519T223809_to_20090520T052608.UTC.thxbin.bz2
|   |-- Modes_dlrop_ul_20090520T052608_to_20090520T072029.UTC.thxbin.bz2
|   |-- ... 9 weitere Dateien ...
|   `-- Modes_dlrop_ul_20090520T214437_to_20090521T050645.UTC.thxbin.bz2
|-- fra
|   |-- Modes_fra_dl_20090519T234156_to_20090520T000256.UTC.thxbin.bz2
|   |-- Modes_fra_dl_20090520T000256_to_20090520T002620.UTC.thxbin.bz2
|   |-- ... 296 weitere Dateien ...
|   |-- Modes_fra_dl_20090520T235003_to_20090521T001500.UTC.thxbin.bz2
|   |-- Modes_fra_ul_20090519T223632_to_20090520T044000.UTC.thxbin.bz2
|   |-- Modes_fra_ul_20090520T044000_to_20090520T060447.UTC.thxbin.bz2
|   |-- ... 11 weitere Dateien ...
|   `-- Modes_fra_ul_20090520T210051_to_20090521T032419.UTC.thxbin.bz2
|-- ko
|   |-- Modes_ko_dl_20090519T235614_to_20090520T005123.UTC.thxbin.bz2
|   |-- Modes_ko_dl_20090520T005123_to_20090520T020400.UTC.thxbin.bz2
|   |-- ... 141 weitere Dateien ...
|   |-- Modes_ko_dl_20090520T235356_to_20090521T012929.UTC.thxbin.bz2
|   |-- Modes_ko_ul_20090519T214802_to_20090520T051138.UTC.thxbin.bz2
|   |-- Modes_ko_ul_20090520T051138_to_20090520T071108.UTC.thxbin.bz2
|   |-- ... 8 weitere Dateien ...
|   `-- Modes_ko_ul_20090520T232902_to_20090521T054328.UTC.thxbin.bz2
```

Listing 6: Verzeichnis mit Mode S-Aufzeichnungen eines Tages auf zentralem Server

Um einem Festplattendefekt vorzubeugen, werden die Daten auf einem RAID1 System gehalten. RAID1 (Automatische Online-Spiegelung der Daten auf einer Zweiten Festplatte [RAID]) schützt zuverlässig vor dem Totalverlust der Daten. Die einfache Redundanz schützt aber nicht vor dem Verlust einzelner kleiner Teile der Daten, da beim Wiederherstellen nach einem Festplattenausfall statistisch mit einem nicht korrigierbaren Lesefehler in einem Bit auf der (jetzt nicht mehr redundanten) Festplatte zu rechnen ist [Harris07]. Dies kann zum Ausfall einer Aufzeichnungsdatei führen, was zu verkraften ist.

8.6.1 Ticketsystem und Cluster

Die Auswertung der Aufzeichnungen soll immer tageweise erfolgen, deshalb werden die Empfangsdateien auch nach Tagen sortiert. Ist ein Tag komplett oder wurde eine maximale Wartezeit von zwei Tagen überschritten, kann mit der Auswertung begonnen werden. Zur automatisierten Verwaltung und Verteilung der Auswertungsaufgaben wurde ein einfaches Ticket-System implementiert.

Ist die Aufzeichnung eines Tages bereit zur Auswertung, wird automatisch ein Ticket für diese Aufzeichnung erstellt und das Ticket in die Ticket-Verwaltung einsortiert. Damit ist es möglich, mehrere Computer an der Auswertung zu beteiligen und zu einem Auswertungscluster zusammenzuschalten. Das ist vor allem dann wichtig, wenn neue Versionen der Auswertesoftware entwickelt werden und alle Aufzeichnungen neu analysiert werden müssen. Die Arbeitsplatzcomputer im Institut binden die Laufwerke des Servers ein, holen sich Tickets vom Server, führen die Analyse durch und schreiben die Ergebnisse zurück auf den Server.

8.6.2 Multimerge – Vereinigung der Empfangsdaten zu einem Datenstrom

Die ACAS-Monitor-Software benötigt **einen** zeitlich geordneten Datenstrom von Mode S-Nachrichten um anhand dieses Datenstroms das Umgebungsmodell mithaltend zu aktualisieren. Nur durch eine zeitliche Vereinigung der Empfangsdatenströme von mehreren Stationen zu einem Datenstrom wird es möglich, das ACAS-Monitoring auch mit einem ganzen Netz von Stationen zu betreiben.

Die Datenauswertung im ACAS-Monitor-Netz muss folglich aus zwei Prozessen bestehen: Erstens Multimerge, das die Empfangsdaten aller Stationen zu einem Datenstrom vereinigt. Und zweitens der ACAS-Monitor-Software (Kapitel 5, 6 und 7), die einen Datenstrom entgegennimmt, Übertragungsfehler ausfiltert, ein Umgebungsmodell aufbaut, ACAS-relevante Dialoge rekonstruiert und mit Flugzeugbahnen zu ACAS-Event-Berichten zusammenstellt.

Jede SPU liefert einen sortierten Datenstrom, der in mehrere Dateien aufgeteilt ist (siehe Listing 6). Durch Aneinanderhängen der Dateien können die Eingangsdatenströme wieder hergestellt werden. Multimerge erledigt das effizient und fügt gleichzeitig die Eingabeströme der Stationen durch „Merging“ zu einem Datenstrom zusammen. Die Funktionsweise des Merging ist folgende:

Multimerge betrachtet von jedem Eingabestrom nur die jeweils erste Nachricht. Die Nachricht mit dem kleinsten Empfangszeitstempel wird als Text auf der Standardausgabe ausgegeben, und der Lese-Zeiger in dem Eingabestrom von dieser Nachricht zur nächsten geschoben. Wieder gibt es für jeden Eingangsstrom eine nächste Nachricht, von diesen wird wiederum die Nachricht mit dem kleinsten Empfangszeitstempel ausgegeben, usw..

Dieser Prozess läuft ab bis alle Nachrichten aus allen Empfangsdateien sortiert ausgegeben wurden. An die Standardausgabe des Multimerge ist direkt die ACAS-Monitor-Software angebunden und verarbeitet die gelieferten Nachrichten.

Da die Datenströme in den Aufzeichnungsdateien bereits sortiert sind, muss Multimerge keine Sortierung, die $O(n \log(n))$ Zeit bei n Nachrichten kostet, vornehmen. Sondern Multimerge kann die Nachrichtenströme, wie oben beschrieben, im zeitlichen Reißverschlussverfahren vereinigen. Diese Vereinigung wird Merging genannt und hat eine Zeitkomplexität von $O(\log k)$ bei k Nachrichten [Knuth2].

8.7 Zusammenfassung des vernetzten ACAS-Monitorings

Das vollständige ACAS-Monitor-Netz hat die in Bild 53 vereinfacht dargestellte Struktur. Das Netz besteht aus fünf Stationen im Feld und einer Zentrale. Die Stationen sind Frankfurt, Korntal,

Oberpfaffenhofen, Braunschweig mit Blick nach Nordwesten und Braunschweig mit Blick nach Südosten.

In jeder Station empfängt eine Thales 1030 MHz-Uplink-SPU und eine Thales 1090 MHz-Downlink-SPU die kompletten Mode S-Aussendungen aller Flugzeuge in Reichweite. Die Software `Thudp2binfile` in dem Stationsrechner jeder Station nimmt die empfangenen Daten entgegen, vervollständigt das Empfangsdatum jeder Nachricht, komprimiert die Daten und speichert sie zwischen. Erst in einer RAM-Disk und bei Bedarf auch für mehrere Wochen auf Festplatte. Sobald die Internetverbindung und die Zentrale bereit sind, überträgt die Software `Uploadrecs.py` die Empfangsdaten verschlüsselt nach Braunschweig. Zur Verschlüsselung und Authentifizierung der Übertragung über das Internet wird das Dateitransferprotokoll SCP des SSH-Paketes benutzt. Die Stationen sind remote-administrierbar. Dabei ist lediglich der Zugriff über SSH auf die Stationsrechner gestattet. Beliebige andere Zugriffsverfahren auf die Stationsrechner und SPUs können bei Bedarf über die verschlüsselte SSH-Verbindung per Tunneling übermittelt werden.

Die Zentrale in Braunschweig nimmt die Aufzeichnungen von den Stationen entgegen und sortiert alle Aufzeichnungen eines Tages zusammen in ein Auswerteverzeichnis. Ein typisches Verzeichnis eines Tages wird ca. 4 GByte groß und umfasst ca. 1000 Aufzeichnungsdateien. Sind die Aufzeichnungen eines Tages komplett, wird automatisch ein Ticket zur Auswertung erstellt. Die Auswertungen werden typischerweise vom Server durchgeführt. Ist der Server ausgelastet, weil z. B. archivierte Aufzeichnungen mit überarbeiteter Software erneut analysiert werden, können weitere Computer die Auswertung unterstützen, indem sie sich ein offenes Ticket suchen und es abarbeiten.

Die Auswertung erfolgt durch zwei gekoppelte Programme: `Multimerge` und die `ACAS-Monitor-Software`. `Multimerge` entpackt die Aufzeichnungen und vereinigt die Datenströme der zehn Empfangsanlagen im zeitlichen „Reißverschlussverfahren“ zu einem Datenstrom, der wie von einer Quelle erscheint. Dieser Datenstrom eines gesamten Tages wird ca. 50 GByte lang, er wird nicht zwischengespeichert, sondern direkt an die `ACAS-Monitor-Software` übergeben.

Die `ACAS-Monitor-Software` kann den Datenstrom nicht komplett am Stück analysieren, sondern betrachtet immer nur eine Nachricht. Die betrachtete Nachricht wird auf Übertragungsfehler untersucht und nur korrekte Nachrichten werden weiterverarbeitet. Die relevanten Informationen aus korrekten Nachrichten werden extrahiert und in ein Umgebungsmodell, die `Plane_DB`, einsortiert oder zu `ACAS-Events` gruppiert (siehe Kapitel 5). Die fertig verarbeitete Nachricht wird aus dem Speicher entfernt. Das Umgebungsmodell und die Liste der `ACAS-Events` werden regelmäßig von nicht mehr relevanten Informationen gesäubert, abgeschlossene `ACAS-Events` werden ausgegeben und weiterverarbeitet.

Damit konnte die `ACAS-Monitor-Software`, die ursprünglich nur für eine Station entwickelt wurde, so erweitert werden, dass beliebig viele Stationen angebunden werden können. Die einzige Begrenzung ist der Datendurchsatz des zentralen Servers und des Auswertecusters.

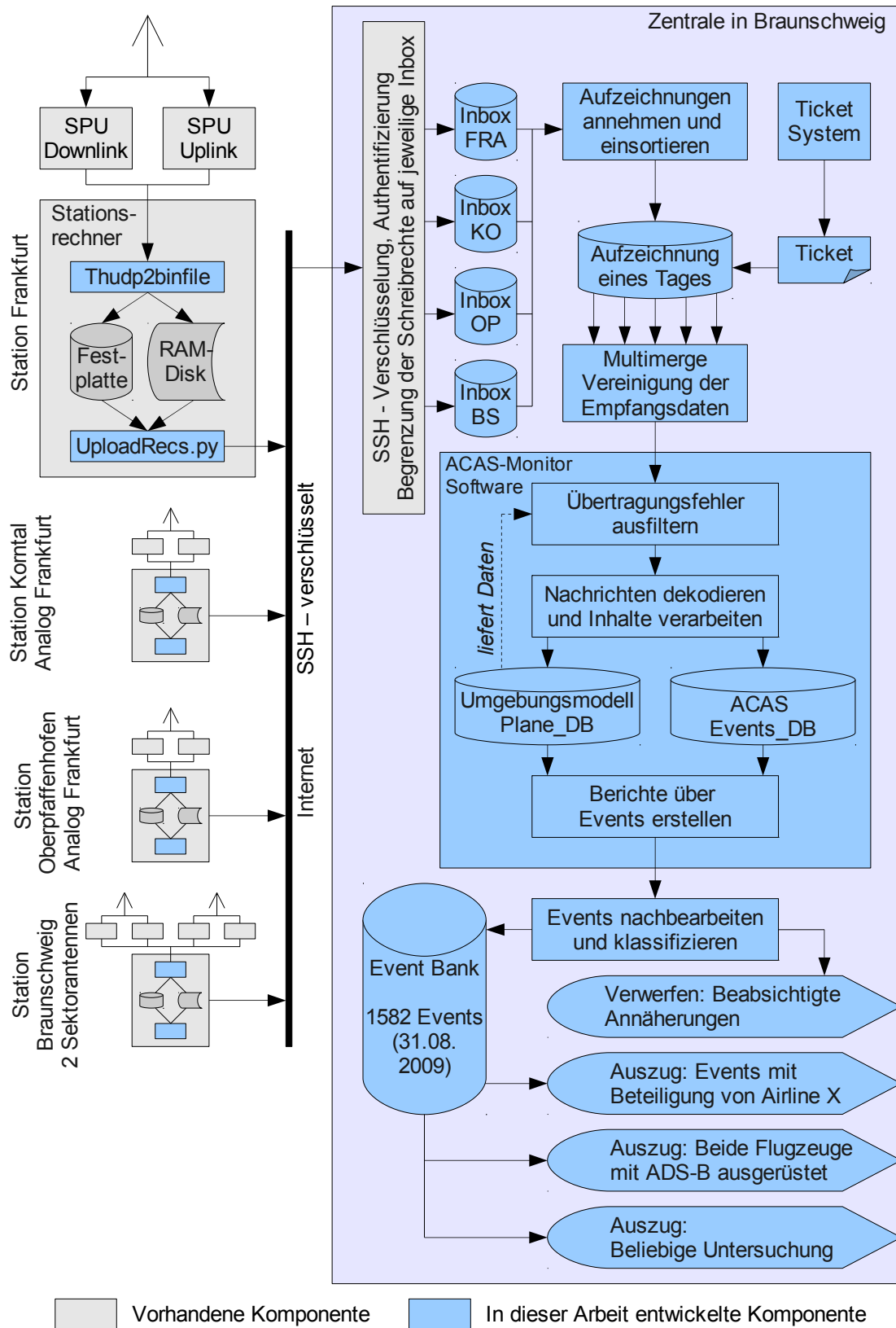


Bild 53: Das ACAS-Monitor-Netz im Überblick

9 Beobachtete ACAS-Events

Seit dem 20. April 2007 wird eine kontinuierliche Dauerauswertung von Mode S-Daten durchgeführt. Zu Beginn mit einer Station auf dem Institutsdach des IfEV in Braunschweig und seit August 2008 mit allen fünf Stationen des ACAS-Monitor-Netzes (Kapitel 8). Die vollständig automatisierte Aufzeichnung und Auswertung dauert bis heute (20. September 2010) an. Täglich analysiert die ACAS-Monitor-Software alle empfangenen Mode S-Nachrichten und baut daraus ein Umgebungsmodell auf (Kapitel 5). Aus den ACAS-relevanten Nachrichten werden die ACAS-Dialoge rekonstruiert (Kapitel 6). Diese rekonstruierten Dialoge und ihre Beschreibung werden als ACAS-Events bezeichnet. Die Event-Berichte konnten während der zweijährigen Daueraufzeichnung ständig weiterentwickelt und verbessert werden, Beispiele für frühe und verbesserte Formen der Berichte finden sich in [Gottst08a], [Gottst09] und [Gottst09a].

In diesem Kapitel wird der Aufbau der Berichte erklärt und es wird ein Überblick über die Beobachtungen gegeben. Mit den dargestellten Events soll die Leistungsfähigkeit des ACAS-Monitorings belegt werden.

9.1 Aufbau der ACAS-Event-Berichte

Die ACAS-Monitor-Software extrahiert aus den Mode S-Aufzeichnungen die ACAS-Events. Ein Modell des ACAS-Events existiert erst einmal nur in der ACAS-Monitor-Software als Datensatz in der ACAS_Events_DB (siehe Kapitel 6). Er umfasst alle zu dem Event empfangenen ACAS-Nachrichten und die Adressen aller beteiligten Flugzeuge. Ist ein Event abgeschlossen, d. h. wurden für eine gewisse Zeit keine neuen Nachrichten zu dem Event gefunden, so wird dieser zusammen mit allen Informationen zu den Flugzeugen aus dem Umgebungsmodell (der Plane_DB) ausgegeben, d. h. auf Datenträger gespeichert.

Jeder Event-Bericht wird als Verzeichnis abgelegt, wodurch Event-Berichte leicht durch Hinzufügen neuer Dateien erweitert werden können. Das Event-Verzeichnis wird nach dem Empfangszeitpunkt der ersten ACAS-Nachricht benannt, z. B. „E_20090323T144751.23456“, Tag und Stunde sind durch den Buchstaben „T“ getrennt. Der Empfangszeitpunkt wird mit einer Auflösung von 10^{-5} s angegeben. Mit dieser „übertriebenen“ Auflösung wird verhindert, dass gleichzeitig ablaufende Events den gleichen Namen bekommen können.

Die ACAS-Monitor-Software erzeugt folgende Dateien:

- E_<Name>.log³⁸: Liste aller RA-relevanten Nachrichten.
- E_<Name>.txt: Alle RA-relevanten Nachrichten als Bitfolge und Erklärung der Bedeutung als leicht lesbarer Klartext.

38 <Name> ist der Name (Empfangszeitpunkt) des jeweiligen Events, z. B. 20090323T144751.23456

- E_<Name>.<Adresse1>.alt³⁹ und E_<Name>.<Adresse2>.alt: Die Höhenprofile der beiden Flugzeuge. Ist nur ein Flugzeug oder sind mehr als zwei Flugzeuge beteiligt, werden entsprechend viele Dateien erzeugt.
- E_<Name>.<Adresse1>.asx und E_<Name>.<Adresse2>.asx: Horizontale Spuren der beteiligten Flugzeuge.

Auf jedem neuen Event werden automatisch eine Reihe von Nachbearbeitungsprogrammen ausgeführt, die das Event-Verzeichnis um etliche Dateien erweitern. Ein typisches Event-Verzeichnis hat nach der Nachbearbeitung folgenden Aufbau:

```
E_20070606T231335.390000
|-- E_20070606T231335.390000.log
|-- E_20070606T231335.390000.txt
|-- E_20070606T231335.390000_4XXXXX.alt
|-- E_20070606T231335.390000_AYYYYY.alt
|-- E_20070606T231335.390000_4XXXXX.asx
|-- E_20070606T231335.390000_AYYYYY.asx
|-- TICKET.postprocess.DONE
|-- E_20070606T231335.390000.alt.png
|-- E_20070606T231335.390000.asx.png
|-- E_20070606T231335.390000.hdist
|-- E_20070606T231335.390000.msg
|-- E_20070606T231335.390000_line_1.msg
|-- E_20070606T231335.390000_line_2.msg
|-- E_20070606T231335.390000.msg.png
|-- alt.gplot
|-- asx.gplot
|-- distpos.txt
|-- msg.gplot
`-- tag.pickle
```

Listing 7: Beispiel für ein Event-Verzeichnis mit Dateien

Die einzelnen Dateien werden im Folgenden erklärt.

Nachbearbeitung von Events

Die Extraktion der Events aus den Mode S-Empfangsdaten und die Nachbearbeitung der Eventberichte (z. B. zur grafischen Darstellung) sind voneinander getrennt in verschiedenen Programmen implementiert. Damit müssen nicht bei jeder Veränderung an der Darstellung die Events neu aus den Aufzeichnungen extrahiert, sondern lediglich die Event-Daten neu verarbeitet werden.

Die einzelnen Nachbearbeitungsschritte sind wiederum in einzelne Programme aufgeteilt, um die Entwicklung zu modularisieren und somit zu vereinfachen. Die Datei TICKET.postprocess.DONE speichert, welches Nachbearbeitungsprogramm in welcher Version das Event bearbeitet hat.

Zu den automatischen Nachbearbeitungsprogrammen zählt zum Beispiel das Entfernen von beabsichtigten Begegnungen (Abschnitt 9.19), das Generieren von Diagrammen und das Zuordnen von Events zu Kategorien wie z. B. „Beide beteiligte Flugzeuge sind mit ADS-B ausgerüstet“.

³⁹ <Adresse1> ist die Mode S-Adresse des Flugzeuges in hexadezimaler Schreibweise: z. B. 3C1234

9.2 ACAS-Event Beispiel 1 – Einfacher Konflikt

Die in diesem Kapitel gezeigten Darstellungen sind mit automatischen Nachbearbeitungsprogrammen erstellt worden. Die realen Events werden unverändert gezeigt, lediglich die Identitäten der Flugzeuge sind entfernt worden, weil die Events willkürlich als Beispiele herangezogen wurden. Die Flugzeuge werden im Folgenden nach den Farben in den Graphen (z. B. Bild 54) als „Rot“ und „Grün“ bezeichnet. Die Farben sind willkürlich gewählt und haben keine Bedeutung.

Das Event-Beispiel 1 hat einen sehr einfachen Verlauf und eignet sich daher sehr gut zur Erklärung der Leistungsfähigkeit der Berichterstattung. An diesem Event werden anschließend alle Diagramme erläutert – in späteren Events wird nur noch auf die für das Beispiel relevanten Diagramme eingegangen.

9.2.1 Höhendigramm

Da ACAS nur vertikale Anweisungen geben kann, ist das Höhendigramm (Bild 54) der Flugzeuge von zentraler Bedeutung und wird bei einer manuellen Analyse von Events meist zuerst betrachtet. Daher wird es auch mit möglichst vielen verdichteten Informationen zum Event angereichert. Das Höhendigramm ist im ACAS-Event-Verzeichnis in der Datei E_20070606T231335.390000.alt.png gespeichert. Um die Grafik zu erzeugen, wird vom Nachbearbeitungsprogramm für jedes Event automatisch ein eigenes Gnuplot-Programm [GNUPlot] generiert, in alt.gplot abgelegt und ausgeführt.

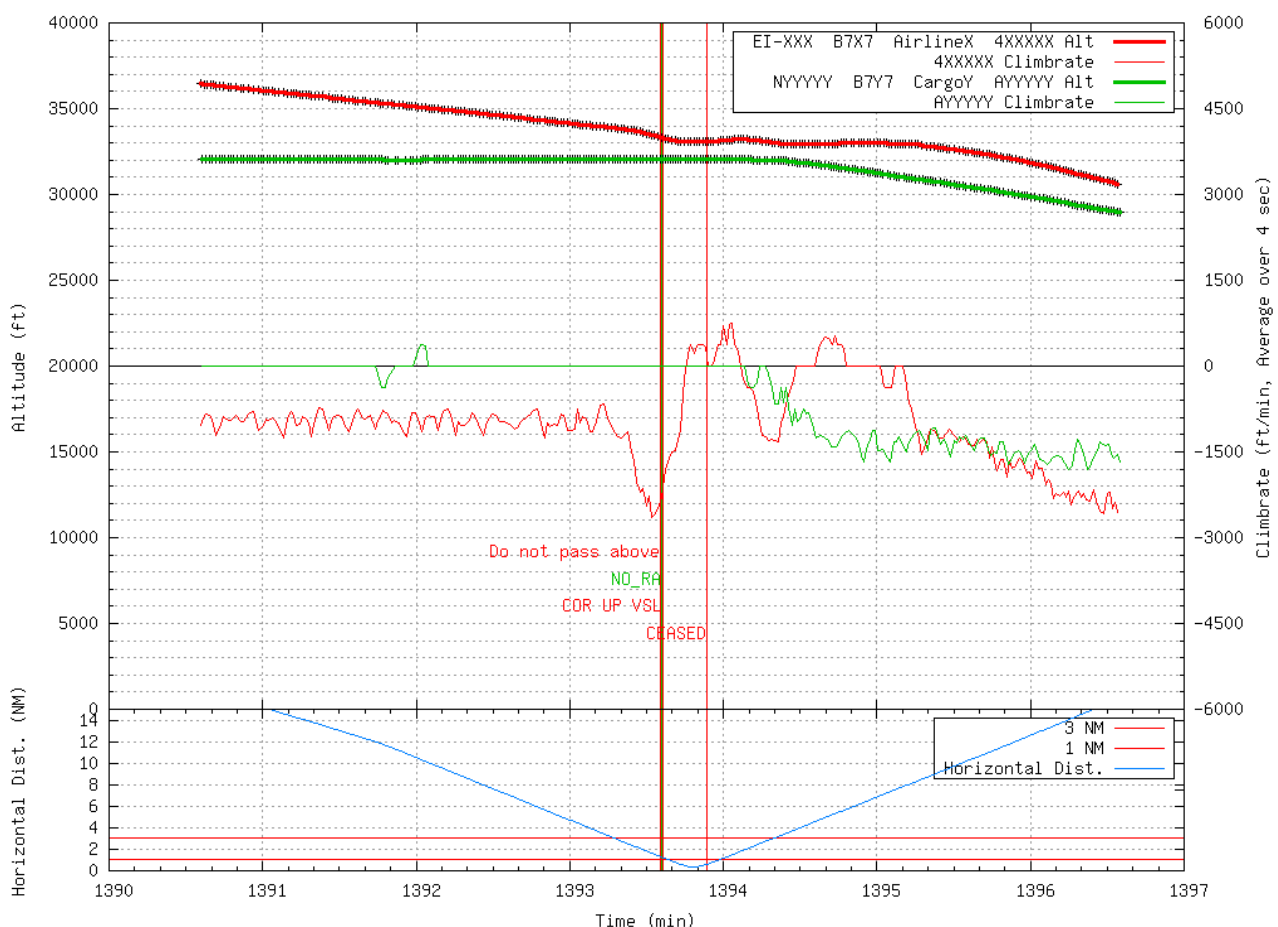


Bild 54: Höhendigramm ACAS-Event Beispiel 1

Im Höhendigramm sind die Druckhöhen⁴⁰ (linke Ordinate in ft) der Flugzeuge als dicke rote bzw. grüne Kurve über der Zeit (Abszisse in Minuten seit Mitternacht) abgetragen, 3 min vor und 3 min nach dem ersten RA. In Bild 54 befindet sich Grün im Reiseflug auf FL 320 (32000 ft) und Rot nähert sich sinkend an Grün an. Jeder Höhenreport wird durch einen kleinen schwarzen Strich auf der Kurve markiert. Bei perfekten Empfangsverhältnissen, wie in diesem Beispiel, werden mehrere Höhenmeldungen pro Sekunde empfangen.

Steigraten

Die präzise Beobachtung des Höhenverlaufs erlaubt, die Ableitungen der Höhen, die Steigraten, zu berechnen. Die Steigraten sind als dünne rote bzw. grüne Kurve auf der rechten Ordinate in ft/min abgetragen. Im linken Teil des Diagramms hat Grün eine Steigrate von null und Rot eine Steigrate von ca. -1100 ft/min.

RA-Meldungen

Die langen senkrechten roten und grünen Geraden markieren das Resolution Advisory. Damit das Bild nicht überladen wird, wurde ein Softwaremodul entwickelt, das den ACAS-Dialog automatisch abstrahiert, d. h. auf wesentliche Informationen reduziert. Eine einzelne ACAS-Kommunikation wird nur dann in das Vertikalprofil eingezeichnet, wenn diese **neue** Informationen zur Koordination oder zu den RAs mitteilt. Zum Empfangszeitpunkt wird die Nachricht mit einer Geraden in der Farbe des Absenders und einem Textlabel eingezeichnet. Dabei spielt es keine Rolle, welche Nachricht (UF16-30, UF16-31 oder DF16-30) diese Information geliefert hat.

Die erste RA-spezifische Information in diesem Event ist eine Koordination: Rot weist Grün das RAC „Do not pass above“ zu. Diese ist durch eine rote Gerade mit dem Text „Do not pass above“ bei 1393 min und 35 s dargestellt. Bereits an dieser Stelle wird deutlich, dass das ACAS-Gerät in Rot angesprochen hat. Mit dem RAC wird die Koordination der ACAS-Systeme beider Flugzeuge vorgenommen. Der ACAS-Computer von Grün speichert das erhaltene RAC und wird es bei der Berechnung eines eigenen RA berücksichtigen, d. h. das ACAS von Grün wird möglichst kein nach oben gerichtetes RA anweisen. Dem Piloten wird diese Koordination nicht angezeigt.

Die unmittelbar folgende grüne Gerade mit dem Text „NO_RA“ zeigt an, dass Grün das RAC akzeptiert und noch kein RA an seinen Piloten ausgegeben hat. Unmittelbar folgte auch eine Meldung von Rot über sein eigenes RA. Der Text an der roten Gerade beschreibt das RA: „COR UP VSL“, dies ist eine Abkürzung für „Corrective Upward Vertical Speed Limit RA“, was den Piloten zum Verringern der Sinkrate auffordert (siehe Abschnitt 2.7.5). Nach ca. 20 s meldet Rot ein „CEASED“, das RA in Rot wurde beendet, der Konflikt somit gelöst.

Manövrierender spricht zuerst an

Das ACAS an Bord von Grün hat bei diesem Event überhaupt nicht angesprochen. Ist ein Flugzeug im Reiseflug, so verringert ACAS die Vorwarnzeit um 3 s. Damit spricht das ACAS im manövrierenden Flugzeug zuerst an und dieses hat die Gelegenheit den Konflikt zu lösen, ohne dass das andere Flugzeug seinen Reiseflug verlassen muss. So verläuft auch dieses Event, das ACAS von Grün muss nicht ansprechen, weil die Reaktion von Rot ausreicht, um den Konflikt zu lösen.

⁴⁰ Referenzdruck 1.013,25 hPa, wie beim SSR üblich

Flugverlauf während des RA

Durch Mithören der DF00-Air-Air-Surveillance-Antworten stehen dem ACAS-Monitoring viele Höhenupdates zur Verfügung. Damit ist zu beobachten, dass Rot kurz vor dem RA noch einmal sein Sinken auf -3000 ft/min verstärkt. Sofort nach dem Ansprechen des ACAS reagiert Rot und beendet seinen Sinkflug innerhalb von ca. 10 s. Nach 20 s ist der Konflikt gelöst. In der ersten Minute nach dem RA werden noch geringfügige Höhenkorrekturen beobachtet, bevor Rot nach Minute 1395 wieder in einen Sinkflug geht. Grün hatte kein RA und hat seinen Reiseflug nicht unterbrochen.

Horizontaler Abstand

Unter dem Höhendigramm ist der horizontale Abstand über der gleichen Zeitachse abgetragen. Er wird aus den ADS-B-Spuren berechnet. Zu Beginn des RA ist der Abstand noch etwas mehr als 1 NM und fällt weiter, bis sich die Flugzeuge passiert haben.

9.2.2 ADS-B-Spuren

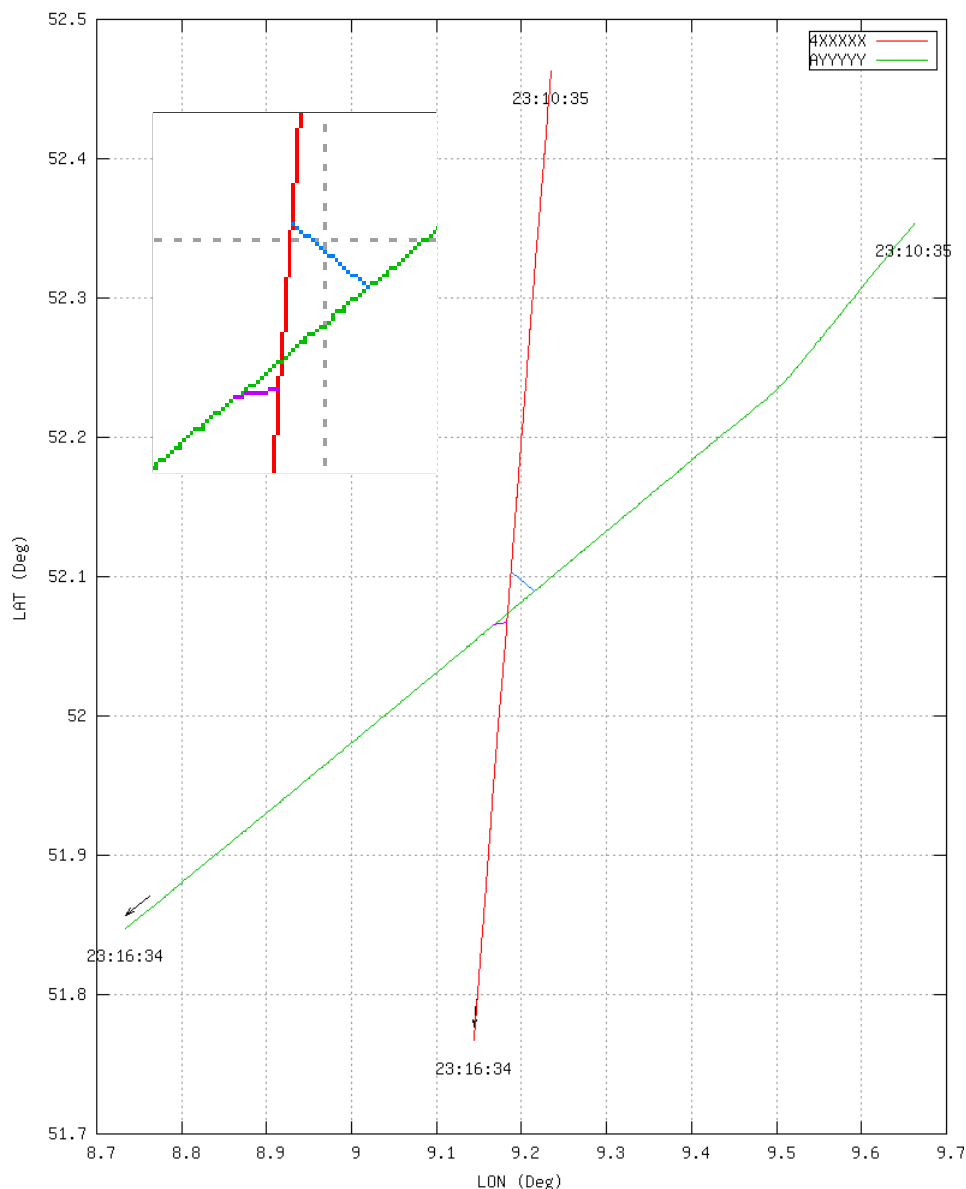


Bild 55: Horizontale Spuren ACAS-Event Beispiel 1

Die ADS-B-Spuren sind in Bild 55 über der geographischen Länge und Breite dargestellt, die Koordinaten sind im WGS84 [WGS84]. Auch diese Graphik wird zu jedem Event automatisch erstellt und ist hier bis auf die Mode S-Adressen unverändert abgebildet.

Die Spuren decken den Zeitraum von 3 min vor und 3 min nach dem RA ab. Die Start- und Endzeitpunkte sind markiert, der Bereich um die Begegnung ist in Bild 55 noch einmal vergrößert dargestellt. Rot fliegt von Norden nach Süden, Grün von Nordosten nach Südwesten. Die blaue Markierung verbindet die Positionen beider Flugzeuge zum Beginn des RA, die violette Markierung verbindet die Positionen am Ende des RA.

Wie bei diesem RA findet bei den meisten RAs kein horizontales Manöver statt. Das ist damit zu erklären, dass ACAS nur vertikale Anweisungen gibt. Lediglich bei RAs in Flughafennahbereichen sind oft auch horizontale Manöver zu beobachten.

9.2.3 ACAS Koordinations-Nachrichten

Im Höhendigramm (Bild 54) kann der RA-Dialog nur abstrahiert dargestellt werden, da schlicht nicht genug Platz ist, um alle Kommunikationen einzuzichnen. Die Grundlage für die eingezeichneten Informationen zum RA stellt der rekonstruierte RA-Dialog (siehe Kapitel 6) dar. Dieser wird im Event-Bericht in den folgenden drei Dateien dargestellt.

```
E_20070606T231335.390000
|-- E_20070606T231335.390000.log      #Liste der ACAS-Nachrichten (Listing 9)
|-- E_20070606T231335.390000.txt     #Klartext (Listing 10)
\-- E_20070606T231335.390000.msg.png #Nachrichtenbild (Bild 56)
```

Listing 8: Dateien mit verschiedenen Darstellungen des ACAS-Dialogs

Liste der ACAS-Nachrichten

Der ACAS-Dialog wird als unkommentierte Liste von Nachrichten im Textformat (siehe Abschnitt 4.1.1) in der Datei E_<NAME>.log (Listing 9) gespeichert. Die Datei wird von der ACAS-Monitor-Software als direkter Auszug aus der ACAS_Events_DB erstellt.

```
# New ACAS-Event: (Nr.    1) 2007-Jun-06 23:13:35.390000
# Involved Aircraft: =>  4XXXXXX  AYYYYY
UF16 Zeit: 83615389.73 ms Nr.: 14590280 Lev: -71 dBm AP: AYYYYY
  » Data: 80800000300800074XXXXXXYYYYYY Timestamp: 1755711858
DF16 Zeit: 83615389.97 ms Nr.: 3573036 Lev: -69 dBm AP: AYYYYY
  » Data: 80E194993000000000000000YYYYYY Timestamp: 1755711873
UF16 Zeit: 83615392.37 ms Nr.: 14590281 Lev: -73 dBm AP: FFFFFFFF
  » Data: 8080000031C00000733XXXXXXXXXX Timestamp: 1755712023
... 28 weitere Nachrichten ...
```

*Listing 9: E_20070606T231335.390000.log – Liste der ACAS-Nachrichten
(„»“ kennzeichnet Zeilenumbrüche, Farben nachträglich eingefügt)*

Die ersten drei Nachrichten waren eine UF16-30-Resolution-Message von 4XXXXXX (Rot) an AYYYYY (Grün), ein DF16-30-Coordination-Reply von AYYYYY (Grün) und ein UF16-31-RA-Broadcast von 4XXXXXX (Rot).

Klartext

Die Liste der ACAS-Nachrichten (Listing 9) stellt nur die rohen Empfangsdaten dar. Um die Inhalte des Dialogs nachzuvollziehen, werden in der Klartextdarstellung die Inhalte zusammen mit ihrer Bedeutung entsprechend dem Standard [ICAO Annex10v4] ausgegeben (Listing 10).

```
New ACAS-Event: (Nr.    1) 2007-Jun-06 23:13:35.390000
Involved Aircraft: =>  4XXXXX (Airline X) (+)  AYYYYY (Cargo-Y) (-)  (Single Threat)

+  UF16  Zeit: 83615389.73 ms  Nr.: 14590280  Lev: -71 dBm  AP: AYYYYY  Data:
  » 80800000300800074XXXXXYYYYYY Timestamp: 1755711858  (Zeit: 23:13:35.390000)
+  UF1630 AP  (AA Receiver): AYYYYY  (Already known in Plane_DB) => Squawk from
  » Plane_DB: YYYY  (Has ACAS)  (Sends ADSB)
+  UF1630 MID (AA Sender):  4XXXXX  (Already known in Plane_DB) => Squawk from
  » Plane_DB: XXXX  (Has ACAS)  (Sends ADSB)
+  MTB: => Interrogating ACAS has one threat.
+  VRC: => Do not pass above.
+  CVC: => No cancellation. New threat.
+  >> VSB-Check of VRC+CVC succeeds.
+  HRC: => No horizontal RAC or no horizontal resolution capability.
+  CHC: => No cancellation or no horizontal resolution capability.
+  >> HSB-Check of HRC+CHC succeeds.
+  >> ACAS MESSAGE DECODING: => CORRECT

-  DF16  Zeit: 83615389.97 ms  Nr.: 3573036  Lev: -69 dBm  AP: AYYYYY  Data:
  » 80E194993000000000000000YYYYYY Timestamp: 1755711873  (Zeit: 23:13:35.390000)
-  DF1630 AP (AA Sender): AYYYYY  (Already known in Plane_DB) => Squawk from Plane_DB:
  » YYYY  (Has ACAS)  (Sends ADSB)
-  Altitude: 32025 ft (+-50 ft)
-  SL:  => ACAS operating at sensitivity level 7 (RA warning time: +35s; altitude band:
  » above FL 200).
-  MTE: => No threat is being processed.
-  ARA: => No vertical RA has been generated.
-  >> ACAS MESSAGE DECODING: => CORRECT

+  UF16  Zeit: 83615392.37 ms  Nr.: 14590281  Lev: -73 dBm  AP: FFFFFFFF  Data:
  » 8080000031C00000733XXXXXXXXX Timestamp: 1755712023  (Zeit: 23:13:35.392000)
+  UF1631 AID (Squawk Sender): X'X'X'X' => UF1631 AID (Unsorted Squawk Sender): XXXX
+  >> Unsorted Squawk from Sender found in Plane_DB: => AA from Plane_DB: 4XXXXX  (Has
  » ACAS)  (Sends ADSB)
+  Altitude: 33300 ft (+-50 ft)
+  MTE: => One threat is being processed.
+  ARA: => RA is corrective.
+  ARA: => Upward sense RA has been generated.
+  ARA: => RA is not increased rate.
+  ARA: => RA is not a sense reversal.
+  ARA: => RA is not altitude crossing.
+  ARA: => RA is vertical speed limit.
+  >> ACAS MESSAGE DECODING: => CORRECT

... 28 weitere Nachrichten ...
```

Listing 10: E_20070606T231335.390000.txt – Klartext des ACAS-Dialoges
(„»“ kennzeichnet Zeilenumbrüche,, Farben nachträglich eingefügt)

Jede Nachricht bildet einen eigenen Absatz im Klartext. Der Absender der Nachricht wird im Text genannt und wird zur besseren Übersicht durch „+“ bzw. „-“ vor der Nachricht gekennzeichnet.

Die erste Nachricht ist eine UF16-30-Resolution-Message von 4XXXXX (Rot) an AYYYYY (Grün). Die Datenbits sind 80800000300800074XXXXXYYYYYY_{hex}, die Bits für Empfänger und Absender sind durch X bzw. Y anonymisiert. In den folgenden Zeilen des Klartextes werden die aktiven Bits, d. h. die Informationen der Nachricht erklärt. Die wichtigste Information der UF16-30 sind die Adressen der beiden Flugzeuge, es wird damit deutlich, dass in Rot ein RA gegen Grün aktiv ist. Das MTB-Bit (Multiple Threat Bit) zeigt an, dass Rot im Moment kein weiteres RA hat. Die VRC-Bits (Vertical RA-Complement) enthalten: „Do not pass above“, was die eigentliche

Koordination darstellt. Damit wird Grün von Rot aufgefördert **nicht oberhalb** von Rot zu passieren.

CVC (Cancel Vertical RA-Complement⁴¹) mit dem Inhalt „No cancellation New threat“ zeigt an, dass mit dieser Nachricht kein vorhergehendes RA-Complement widerrufen wird. Im VSB-Feld (Vertical Sense Bits) ist noch eine Prüfsumme für VRC und CVC, diese ist in dieser Nachricht korrekt. HRC-, CHC- und HSB-Bits sind für horizontale Ausweichenweisungen von ACAS III vorbehalten und in dieser Nachricht korrekter Weise alle 0. Am Ende des Abschnitts der Nachricht wird mit „ACAS MESSAGE DECODING: => CORRECT“ noch angegeben, ob die Bits der Nachricht konform zum [ICAO Annex10v4] waren oder logische Fehler enthielten.

In Listing 10 sind noch zwei weitere Nachrichten abgedruckt, insgesamt hatte das Event 31 Nachrichten. Die Nachrichten sind ähnlich aufgebaut wie die oben erläuterte. Der Klartext erleichtert die Betrachtung der RA-relevanten Nachrichten gegenüber den rohen Mode S-Daten.

Nachrichtendiagramm

Wesentlich schneller als der Klartext ist das Nachrichtendiagramm zu lesen. In ihm sind alle 31 Nachrichten mit den wesentlichen Inhalten abgebildet. Das Nachrichtendiagramm wurde gemeinsam mit Dorothee Schröder [Schröder09] im Rahmen einer Studienarbeit entwickelt.

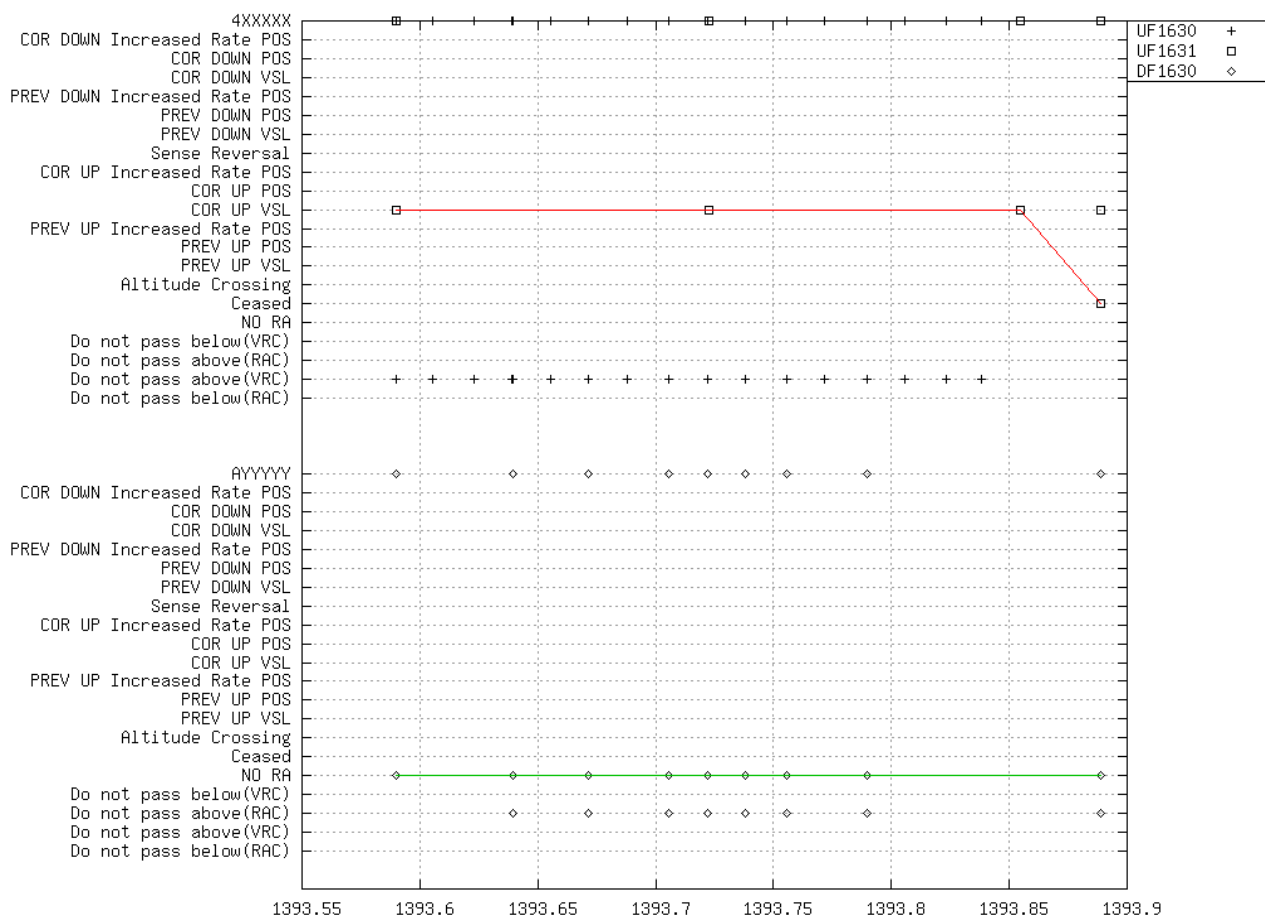


Bild 56: Nachrichtendiagramm

41 Das „Cancel Vertical RA-Complement“-Feld ist im [ICAO Annex10v4] tatsächlich mit CVC benannt.

Jeder Nachrichtentyp hat im Nachrichtendiagramm sein eigenes Symbol, die UF16-30-Resolution-Message hat +, UF16-31-RA-Broadcast □ und DF16-30-Coordination-Reply ◇. Über der Zeit (Minuten seit Mitternacht) sind in der oberen Hälfte die Nachrichten, die von 4XXXXX gesendet wurden, dargestellt, in der unteren Hälfte die von AYYYYY. In den Zeilen neben den Adressen sind alle Nachrichten zu ihren Empfangszeitpunkten eingetragen. In den Zeilen darunter sind die Inhalte der Nachrichten markiert.

4XXXXX sendet sekundlich UF16-30-Resolution-Messages mit dem VRC „Do not pass above“ zur Koordination an AYYYYY. Zu Beginn des RAs und dann alle 8 s sendet 4XXXXX einen UF16-31-RA-Broadcast mit dem RA „Corrective Upward Vertical Speed Limit“.

Die UF16-30 von 4XXXXX beantwortet das Flugzeug AYYYYY mit DF16-30-Coordination-Replies. Ab der zweiten DF16-30 bestätigt AYYYYY auch das zugewiesene RAC. AYYYYY meldet in den DF16-30, dass kein RA vorliegt. Am Ende des Dialoges meldet 4XXXXX in einer UF16-31, dass der Konflikt gelöst wurde mit einem „Ceased“.

Die RA-Meldungen der Flugzeuge sind mit Linien in den entsprechenden Farben verbunden. Nicht standard-konforme Wechsel zwischen verschiedenen RAs werden so schnell deutlich (siehe auch Event Beispiel 17 in Abschnitt 9.18).

9.3 ACAS-Event Beispiel 2 – Abschwächung des RA nach prompter Reaktion

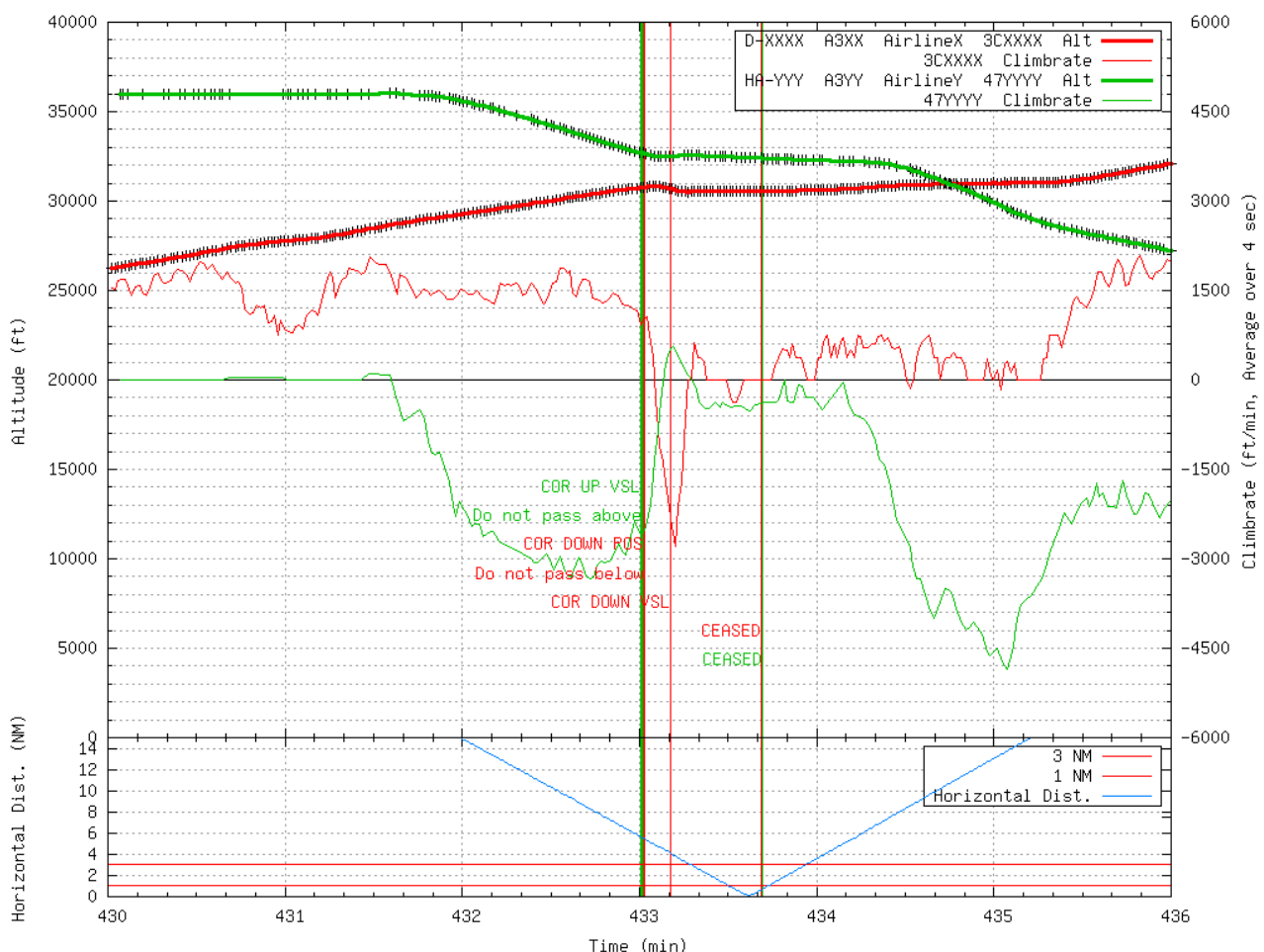


Bild 57: Höhendigramm ACAS-Event Beispiel 2 – Abschwächung des RA

Grün befindet sich im Sinkflug mit ca. 3000 ft/min, Rot im Steigflug mit 1500 ft/min. Als beide nur noch ca. 1500 ft vertikalen Abstand haben, spricht ACAS in beiden Flugzeugen an. Grün erhält ein Corrective Upward Vertical Speed Limit RA, welches prompt befolgt wird. Grün fängt innerhalb von 10 s ab. Rot erhält ein stärkeres RA: Corrective Downward Positive, welches nicht nur zum Abfangen, sondern zum Sinkflug auffordert. Rot reagiert prompt und geht innerhalb von 10 s aus dem Steigflug in einen Sinkflug mit fast 3000 ft/min über.

ACAS überprüft sekundlich, ob das RA noch der sich verändernden Verkehrssituation entspricht. Wegen der prompten Reaktion kann ACAS schon nach ca. 10 s das RA in Rot abschwächen und weist nur noch Abfangen (COR DOWN VSL) an. Im Höhendiagramm werden nur neue Informationen eingetragen. Das abgeschwächte RA ist eine neue Information und wird durch eine senkrechte rote Gerade mit entsprechender Beschriftung markiert. Auch dieses RA wird prompt befolgt: Rot gibt seinen gerade erst eingenommenen Sinkflug auf und fängt ab.

Zu Beginn des RA ist der horizontale Abstand der Flugzeuge noch 6 NM, während des RA fällt er auf fast null. Nach insgesamt ca. 40 s ist der Konflikt gelöst.

Da die ACAS-Geräte in beiden Flugzeugen angesprochen haben, sind im Nachrichtendiagramm (Bild 58) wesentlich mehr Nachrichten eingezeichnet als in Beispiel Event 1. Die Abschwächung des RA ist deutlich als Stufe zu erkennen.

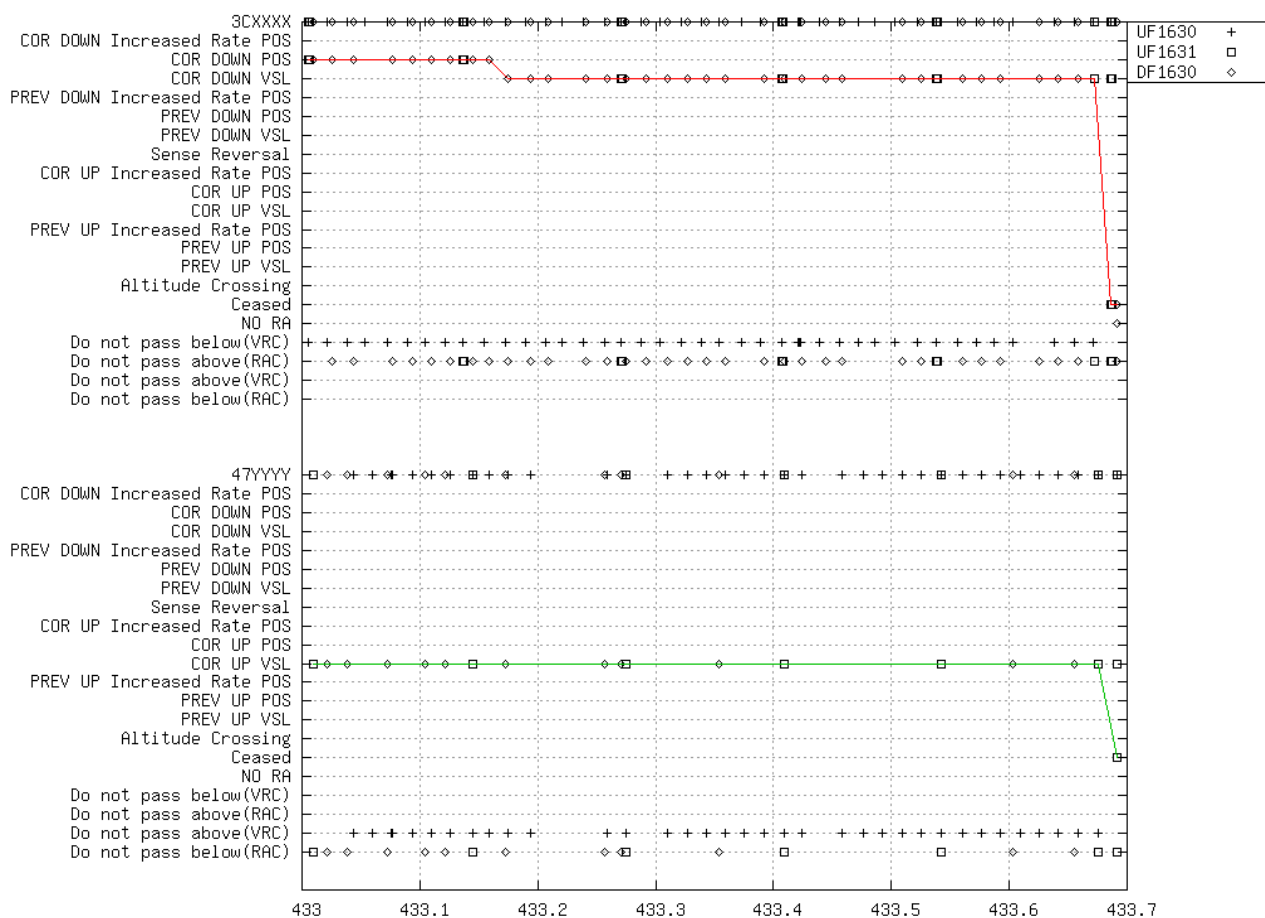


Bild 58: Nachrichtendiagramm – Abschwächung des RA

Bild 59 zeigt, dass auch in diesem Event die Flugzeuge keine horizontalen Manöver durchführen. Wie bereits erwähnt entspricht dies den Erwartungen, da ACAS nur vertikale Ausweichanweisungen gibt.

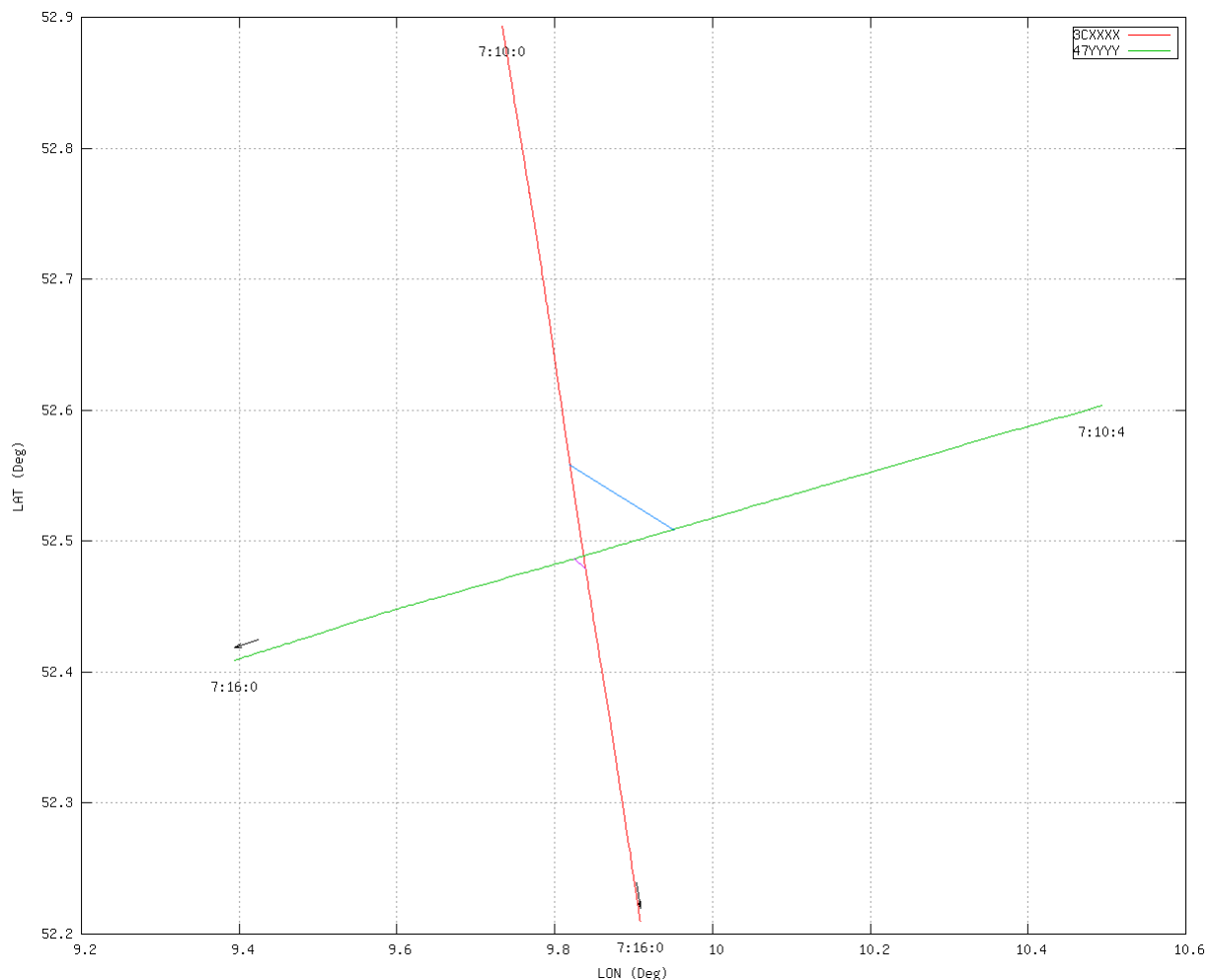


Bild 59: Horizontale Spuren ACAS-Event Beispiel 2

Um die Lesbarkeit der folgenden Abschnitte zu erhöhen, sind die Events und die dazugehörigen Grafiken auf benachbarten Doppelseiten angeordnet. Leider entstehen dabei zwangsläufig leere Bereiche auf den Seiten.

9.4 ACAS-Event Beispiel 3 – Hohe vertikale Rate

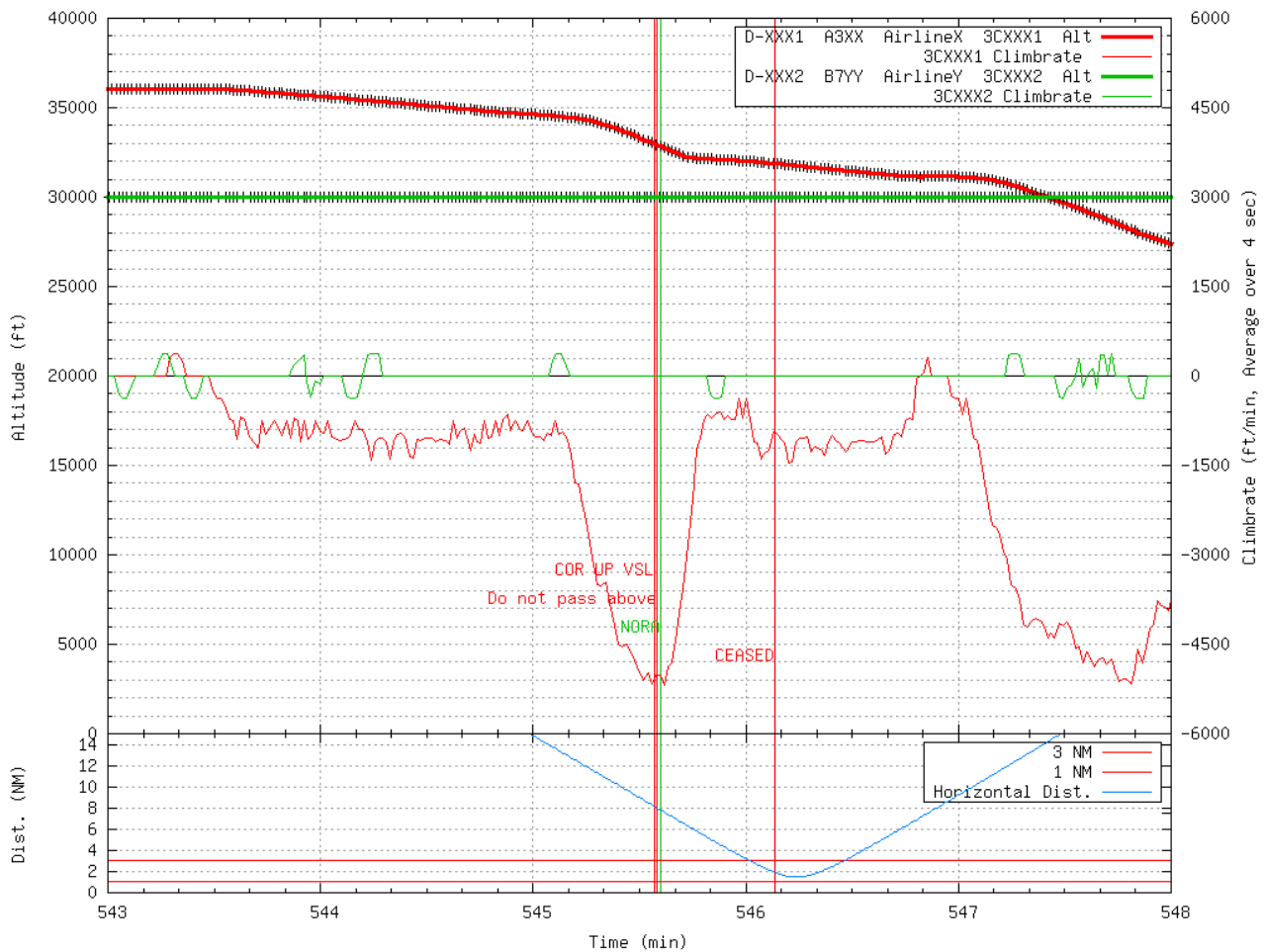


Bild 60: Höhendigramm ACAS-Event Beispiel 3 – Hohe vertikale Rate

In Bild 60 spricht ACAS an, obwohl beide Flugzeuge immer noch 3000 ft vertikal und 8 NM horizontal voneinander entfernt sind. Der Grund ist, dass ACAS in erster Linie nicht Abstände, sondern Vorwarnzeiten prüft. Ab 30000 ft Höhe beträgt die Vorwarnzeit 35 s. Rot sinkt immerhin mit ca. 5000 ft/min. Sein ACAS spricht dementsprechend früh, wie im Beispiel, bei fast 3000 ft Höhendifferenz an, weil diese 3000 ft bei 5000 ft/min in 36 s durchfliegen wären (Vertikaler Test, Abschnitt 2.1). Der Verlauf des horizontalen Abstands unten in Bild 60 zeigt auch, dass die verbleibenden 8 NM Schrägabstand in den nächsten ca. 35 s schwinden werden (Horizontaler Test). Bei Ansprechen beider Tests spricht ACAS an und generiert ein RA.

Da sich Grün im Reiseflug befindet, verringert ACAS seine Vorwarnzeit um 3 s, wodurch der Manövrierende die Gelegenheit erhält, ein RA zu wählen und sein Manöver abubrechen. Im obigen Beispiel genügt die prompte Reaktion von Rot, um den Konflikt zu lösen, das ACAS von Grün muss nicht aktiv werden. Derart hohe vertikale Raten sind keine Seltenheit. Die höchste beobachtete vertikale Rate beim Auslösen eines RA waren 12.000 ft/min Steigrate bei einem Airbus A320, vermutlich ein Testflug von Airbus.

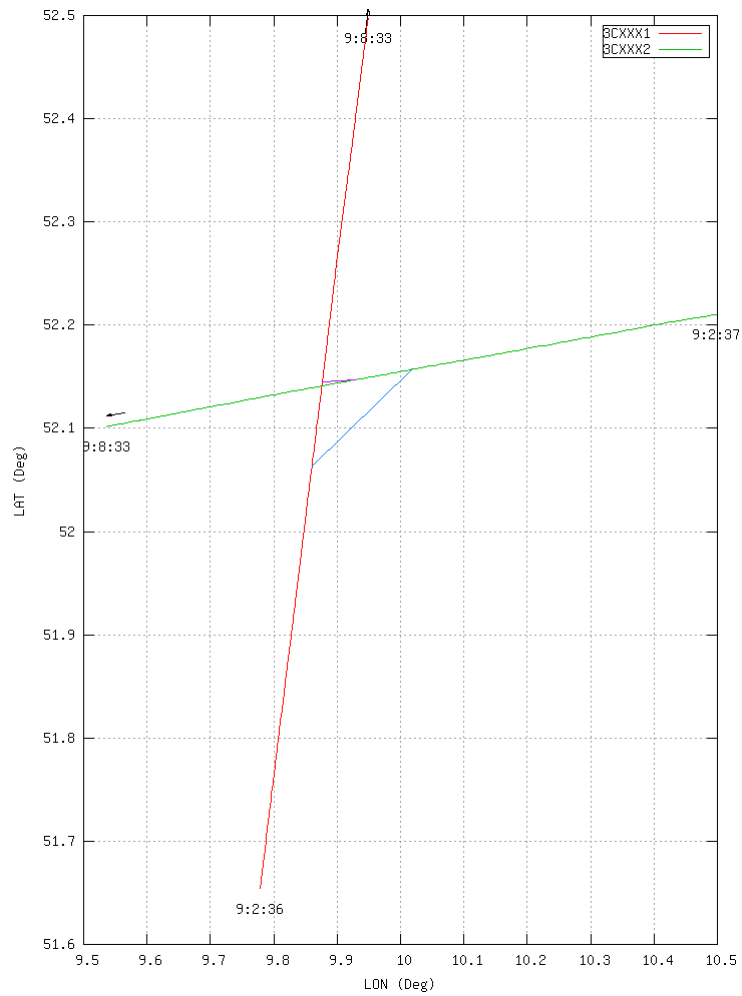


Bild 61: Horizontale Spuren ACAS-Event Beispiel 3

Bild 61 zeigt die horizontalen Spuren der Flugzeuge, es werden keine horizontalen Manöver durchgeführt.

9.5 ACAS-Event Beispiel 4 – Ungenaue Höhenhaltung

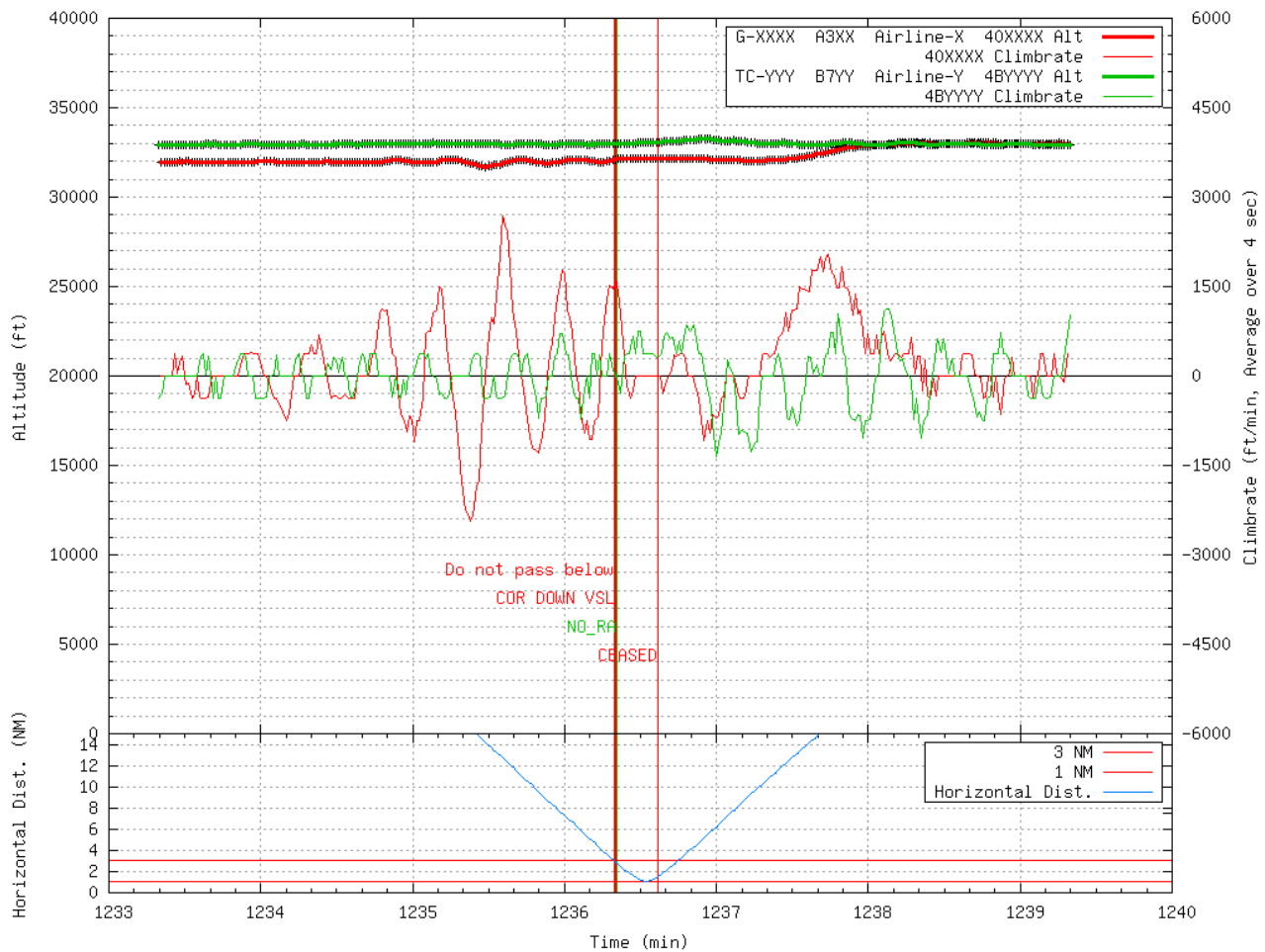


Bild 62: Höhendigramm ACAS-Event Beispiel 4 – Ungenaue Höhenhaltung

Bild 62 und Bild 63 zeigen eine Begegnung, bei der die Flugzeuge 1000 ft vertikal separiert sind. Eine Minute vor der Begegnung ist die ungenaue Höhenhaltung insbesondere von Rot zu beobachten. Das ständige Sinken und Steigen führt letztlich auch zum Ansprechen von ACAS in dieser Begegnung.

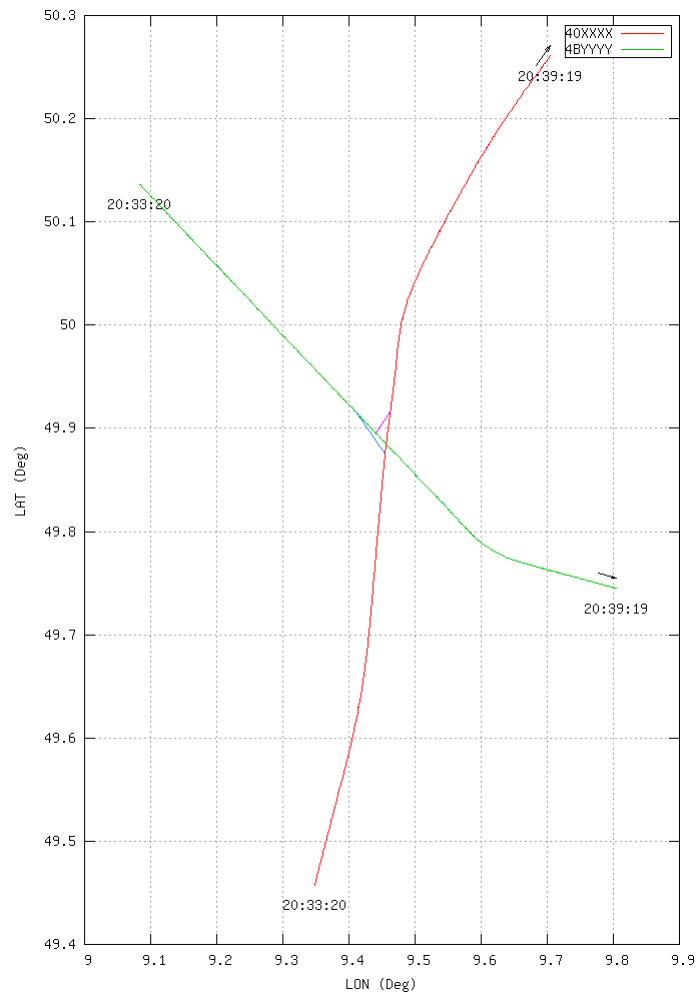


Bild 63: Horizontale Spuren ACAS-Event Beispiel 4

9.6 ACAS-Event Beispiel 5 – Flugzeuge fliegen übereinander, oberes sinkt zuerst

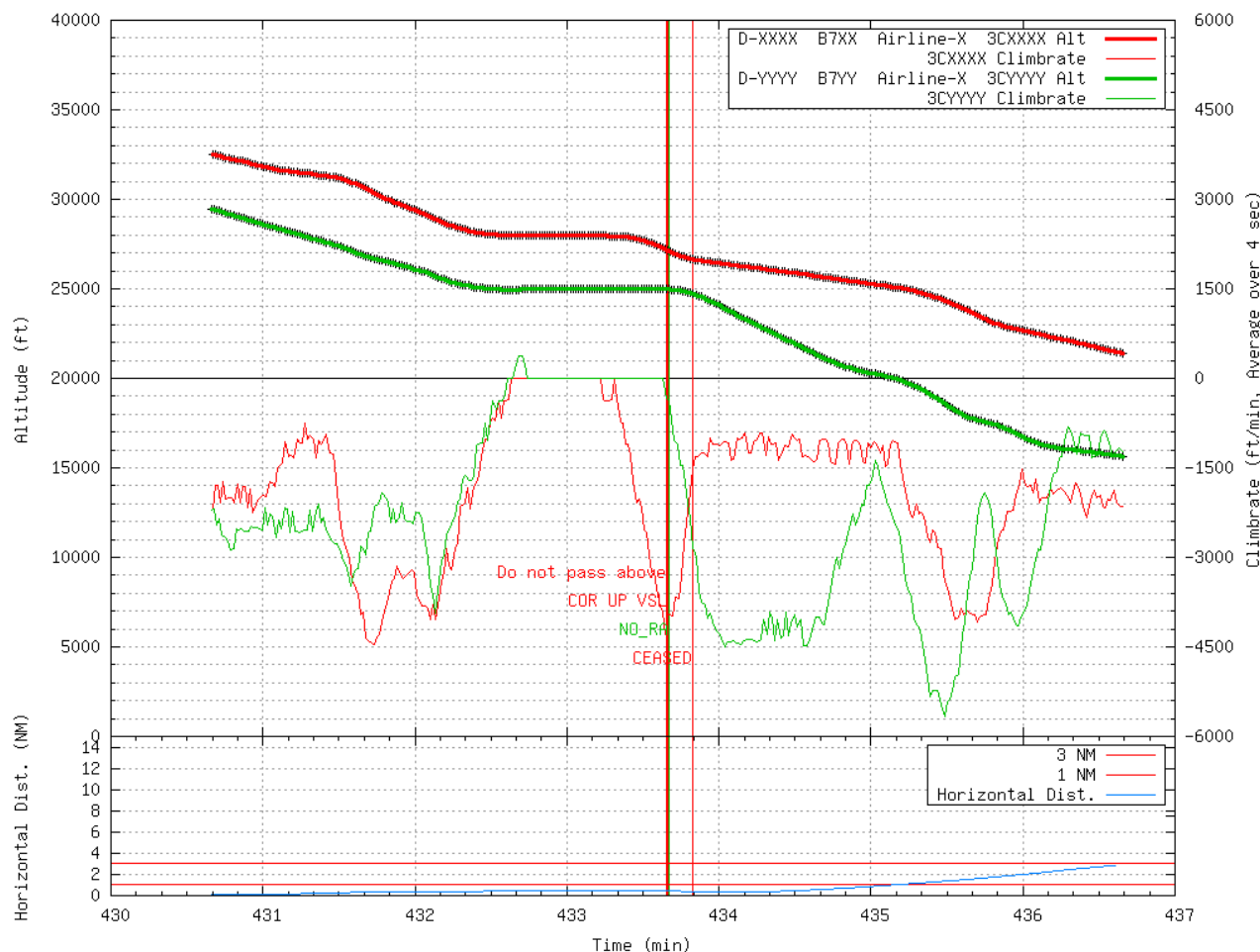


Bild 64: Höhendigramm ACAS-Event Beispiel 5 – Rot fliegt über Grün und sinkt zuerst

In diesem Event fliegen die beteiligten Flugzeuge der gleichen Airline direkt übereinander. Das zeigt sich in Bild 65 und beim horizontalen Abstand in Bild 64. Der obere (Rot) beginnt als erster zu sinken. Nach 20 s hat er die Sinkrate von 4200 ft/min eingenommen. Das ACAS von Rot spricht an und Rot verringert seinen Sinkflug. Da Grün jetzt auch sinkt ist der Konflikt bereits nach 10 s gelöst.

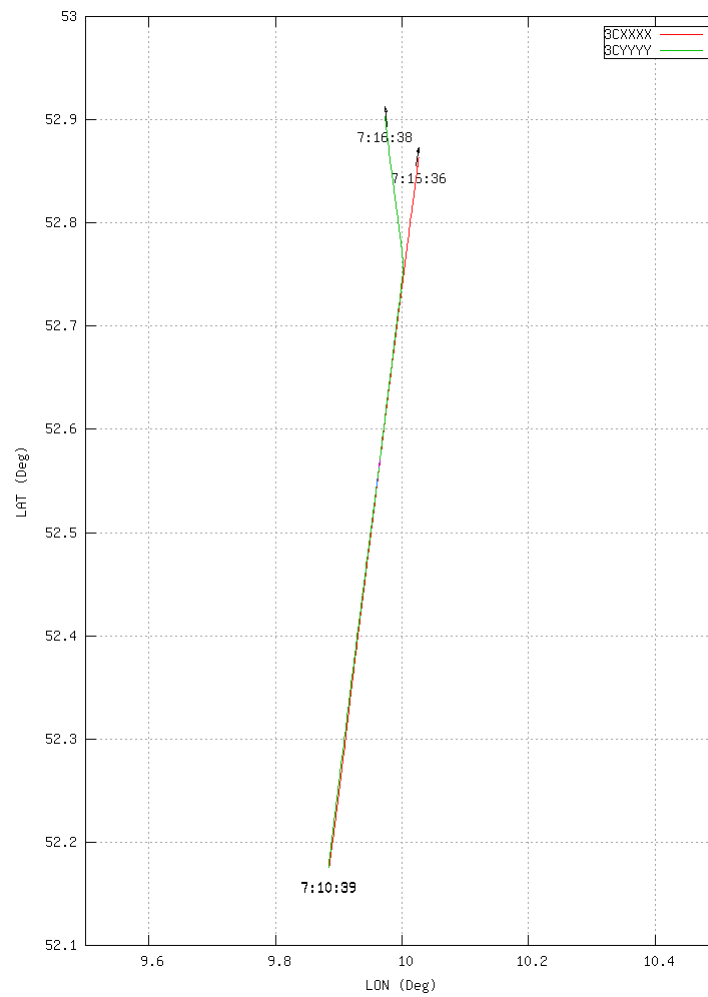


Bild 65: Horizontale Spuren ACAS-Event Beispiel 5

9.7 ACAS-Event Beispiel 6 – Startendes Flugzeug in Konflikt mit landendem

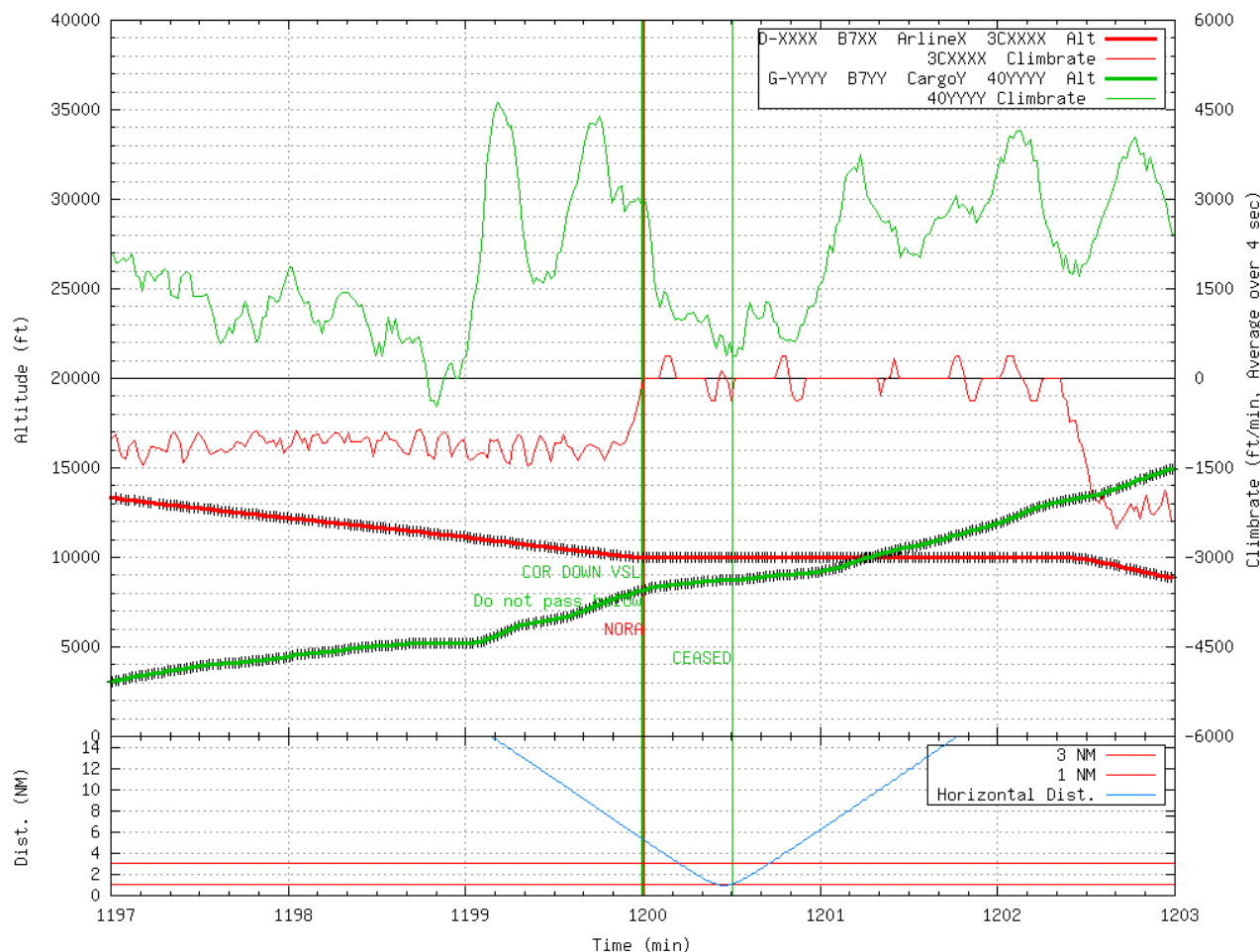


Bild 66: Höhendigramm Beispiel Event 6 in TMA

Bild 66 zeigt das Höhendigramm eines Events in einer TMA (Terminal Maneuvering Area, Nahverkehrsbereich). Das ACAS von Grün weist ein Corrective Downward Vertical Speed Limit an, was prompt befolgt wird. Da Rot sich (seit wenigen Sekunden) im Level Flight befindet, verzögert das ACAS in Rot sein RA und muss schließlich gar kein RA anweisen. Nach 30 s ist der Konflikt gelöst. Bei Events in einer TMA sind auch horizontale Manöver, die wie in Bild 67 über einfache Kursänderungen hinausgehen, zu beobachten.

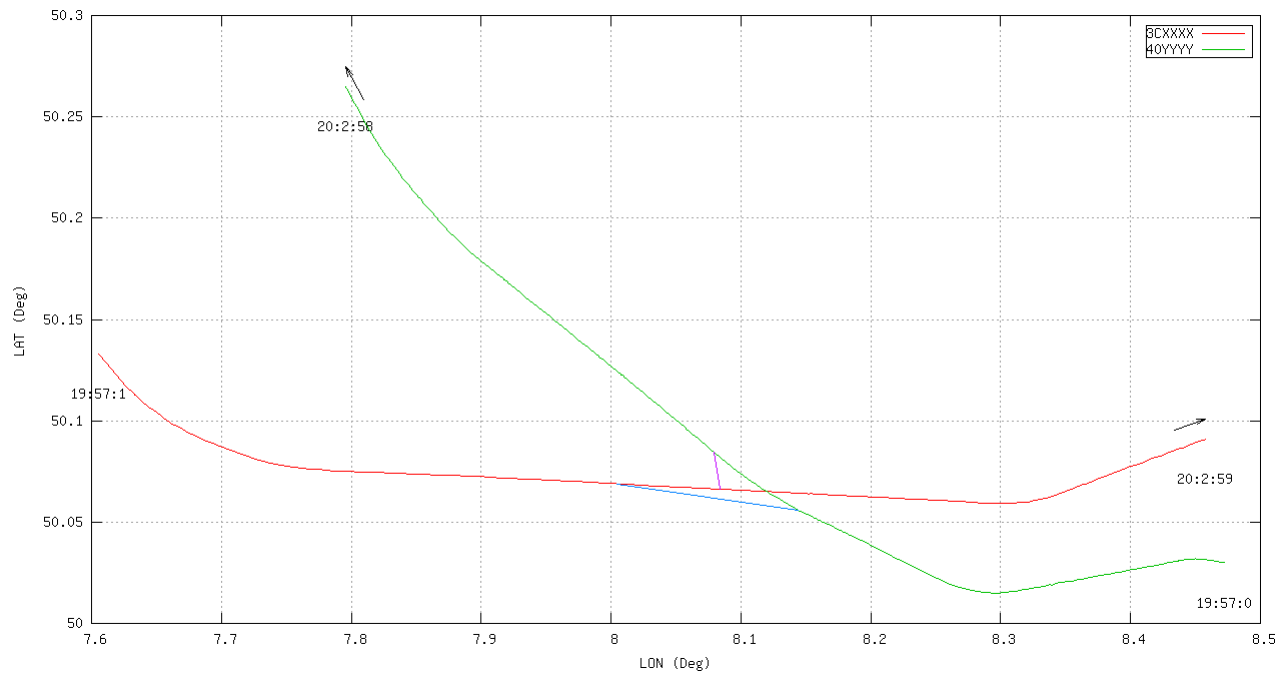


Bild 67: Horizontale Spuren ACAS-Event Beispiel 6 in TMA

9.8 ACAS-Event Beispiel 7 – Einschwenken auf Doppelbahnsystem

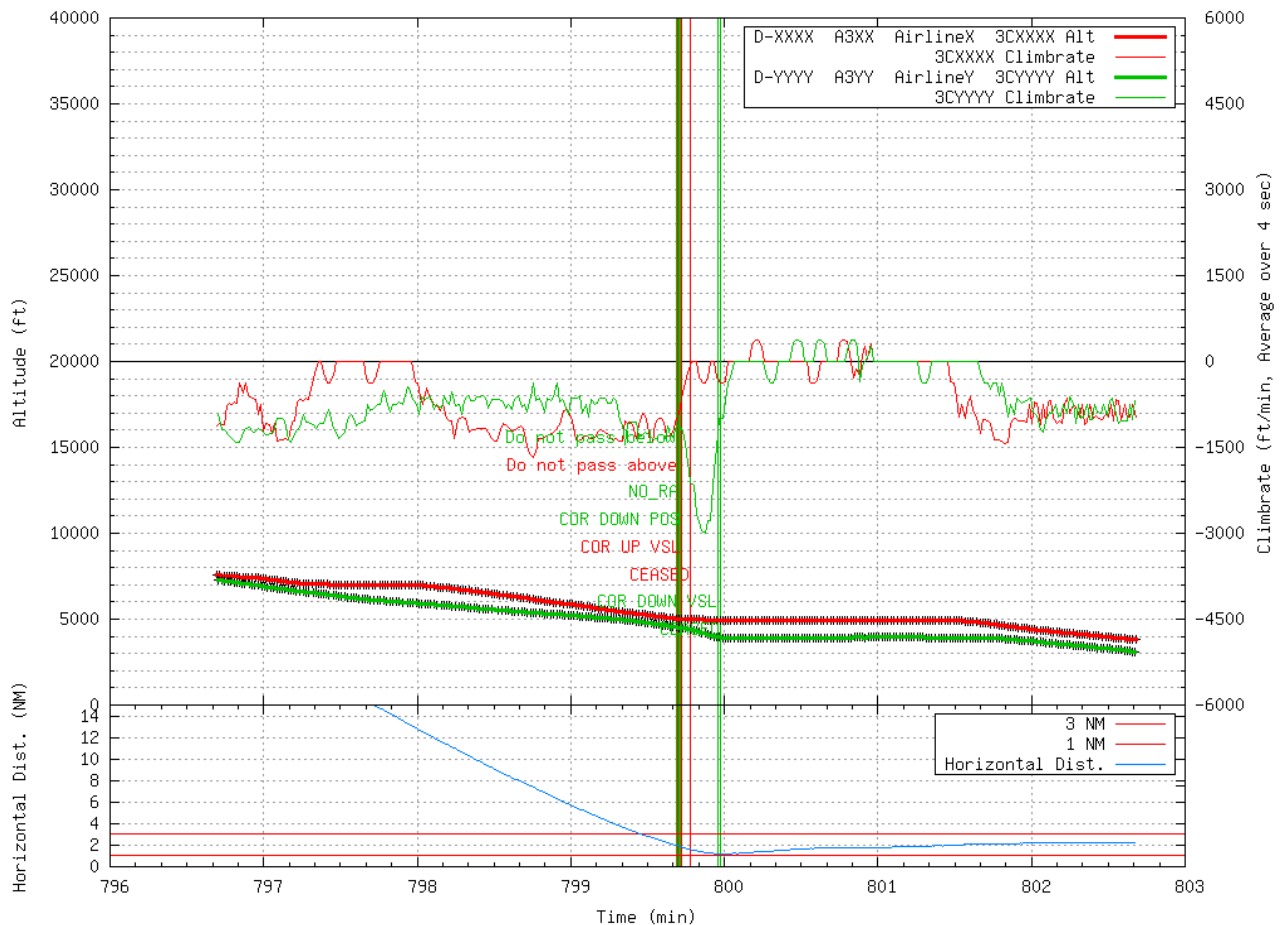


Bild 68: Höhendigramm ACAS-Event Beispiel 7 – Einschwenken auf Landebahnen

Beim Einschwenken auf das Doppelbahnsystem von Frankfurt kommt es regelmäßig zu ACAS-Events wie in den Bildern 68 und 69. Diese stellen einen Stressfaktor für Piloten dar, und Ausweichmanöver unterbrechen den normalen Verkehrsablauf. ACAS-Monitoring dokumentiert solche Events automatisch. Im Beispiel muss Rot abfangen und Grün verstärkt sinken.

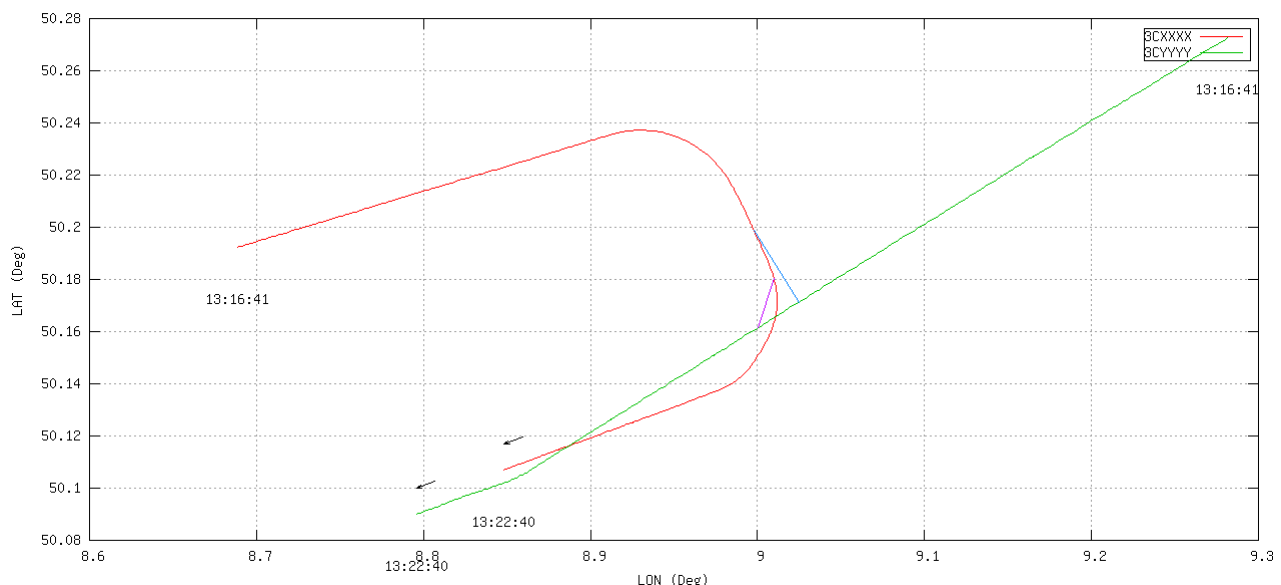


Bild 69: Horizontale Spuren ACAS-Event Beispiel 7 – Einschwenken auf Landebahnen

9.9 ACAS-Event Beispiel 8 – Preventive RA

Die bisher gezeigten Events 1 bis 7 thematisierten das Zustandekommen des ACAS-Eingriffs. Die folgenden Events 8 bis 10 sind Beispiele für verschiedene Resolution Advisories. Die bisher gezeigten RAs waren alle Corrective, d. h. sie haben den Piloten zum Ändern der vertikalen Rate aufgefordert. Bei einem Preventive RA wird dagegen lediglich das Einnehmen einer bestimmten vertikalen Rate von ACAS verboten. Der Pilot wird angehalten, seine vertikale Rate nicht zu verändern. In Bild 70 wird Rot aufgefordert nicht zu steigen, sondern weiter im Reiseflug zu bleiben.

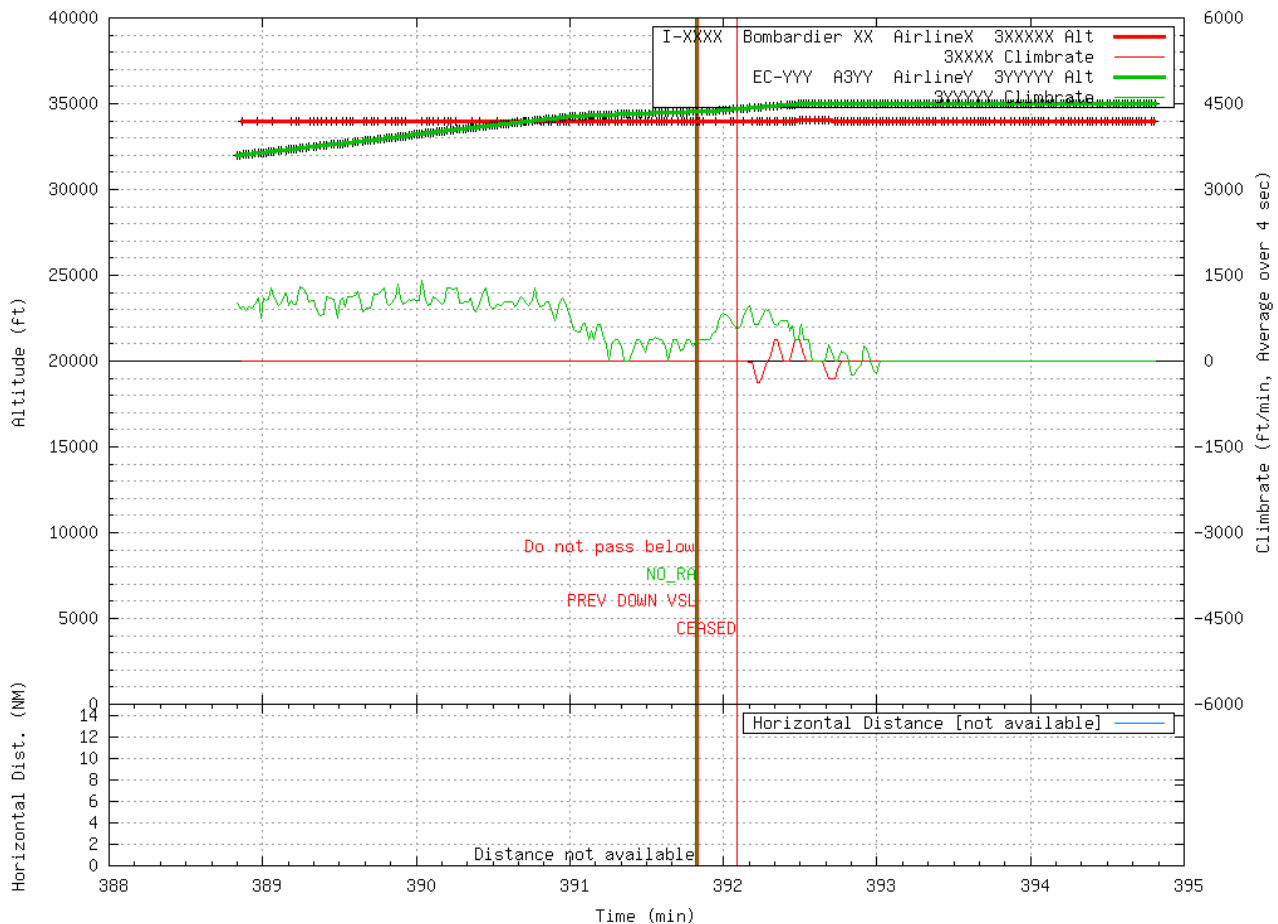


Bild 70: Höhendigramm ACAS-Event Beispiel 8 mit Preventive RA

Obwohl Preventive RAs einen schwächeren Eingriff als Corrective RAs darstellen, können sie trotzdem seltener beobachtet werden. Mit Stand 20. Juli 2009 wurde kein Event mit einem Preventive RA beobachtet, in dem beide Flugzeuge mit ADS-B ausgerüstet waren und das Preventive RA im Verlauf des Events nicht zu einem Corrective RA verstärkt wurde. Darum wird hier ein Event gezeigt, in dem nur ein Flugzeug ADS-B aussendet. Das Höhendigramm enthält demzufolge keinen horizontalen Abstand und im horizontalen Bild (Bild 71) ist nur eine Spur dargestellt. Beginn und Ende der RA sind mit Kreuzen markiert.

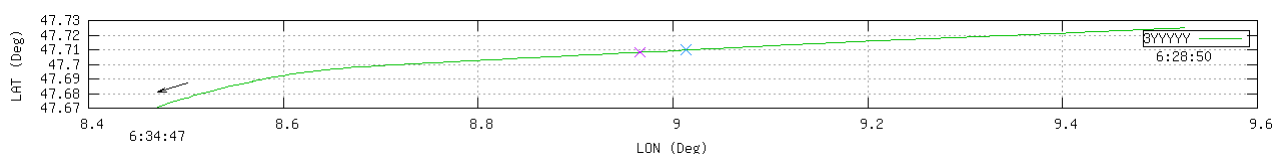


Bild 71: Horizontale Spur des mit ADS-B ausgerüsteten Flugzeuges in ACAS-Event Beispiel 8

9.10 ACAS-Event Beispiel 9 – Altitude Crossing RA

Weist ACAS ein Ausweichmanöver an, bei dem sich die Höhen beider Flugzeuge kreuzen, so wird das als Altitude Crossing RA bezeichnet. Altitude Crossing RAs zwingen den Piloten für eine gewisse Zeit zur Bedrohung hin zu manövrieren und stellen somit eine besondere Belastung dar. Auch diese Events werden vom ACAS-Monitor minutiös dokumentiert.

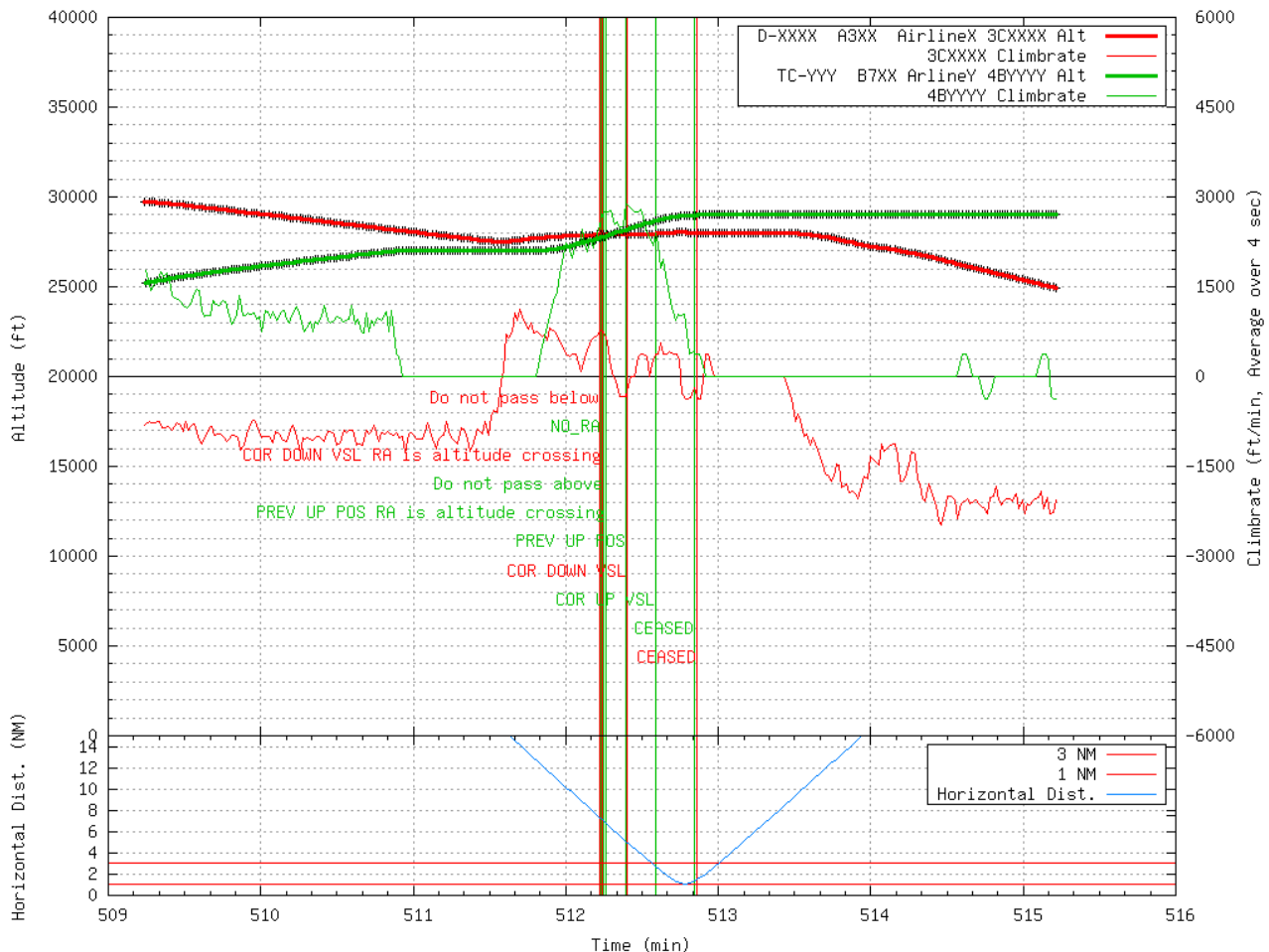


Bild 72: Höhendigramm ACAS-Event Beispiel 9 – Altitude Crossing

Bild 72 zeigt das automatisch erstellte Höhendigramm eines Altitude Crossing RA. Die Darstellung der Kommunikation im Höhendigramm ist zwar auf geänderte Informationen beschränkt, dennoch ist die Kommunikation hier nicht mehr mit einem Blick zu erfassen. Zur Lesbarkeit sei noch einmal darauf hingewiesen: Die Beschriftungen sind immer links an den dazugehörigen senkrechten Geraden angefügt; die Beschriftungen sind in der Farbe der entsprechenden Geraden und laufen von oben nach unten in der Reihenfolge wie die Geraden von links nach rechts laufen.

Der Kommunikationsablauf, wie er im Höhendigramm dargestellt ist, wird nun erklärt: Rot weist Grün das RA-Complement „Do not pass below“ (dt: „Passiere nicht unterhalb“) zu. Grün meldet mit „NO_RA“ noch kein RA zu haben. Rot gibt ein Corrective Downward Vertical Speed Limit RA mit Altitude Crossing, im Bild mit „COR DOWN VSL RA is altitude crossing“ abgekürzt (dt: „Abfangen und Gegner passieren lassen“), an seinen Piloten. Im gleichen Augenblick berechnet Grün ein eigenes RA und beginnt es mit Rot zu koordinieren, d. h. Grün weist Rot das RAC „Do not pass above“ zu. Diese vier Informationen werden nacheinander, fast gleichzeitig ausgetauscht.

Augenblicke später hört das ACAS-Monitoring auch die genaue Beschreibung des RA in Grün: Preventive Upward Positive Altitude Crossing (dt: „Weiter steigen, kreuze die Höhe des Gegners“).

Nach ca. 10 s findet die Höhenkreuzung statt und die ACAS-Geräte können das Altitude Crossing fallen lassen. Grün meldet jetzt ein Preventive Upward Positive RA und Rot ein Corrective Downward Vertical Speed Limit RA. Nach weiteren ca. 15 s können die RAs ganz beendet werden.

Bemerkenswert an diesem Event ist außerdem, dass bei 512 min und 34 s, der horizontale Abstand unter 3 NM sinkt und die Flugzeuge vertikal nur 625 ft separiert sind.

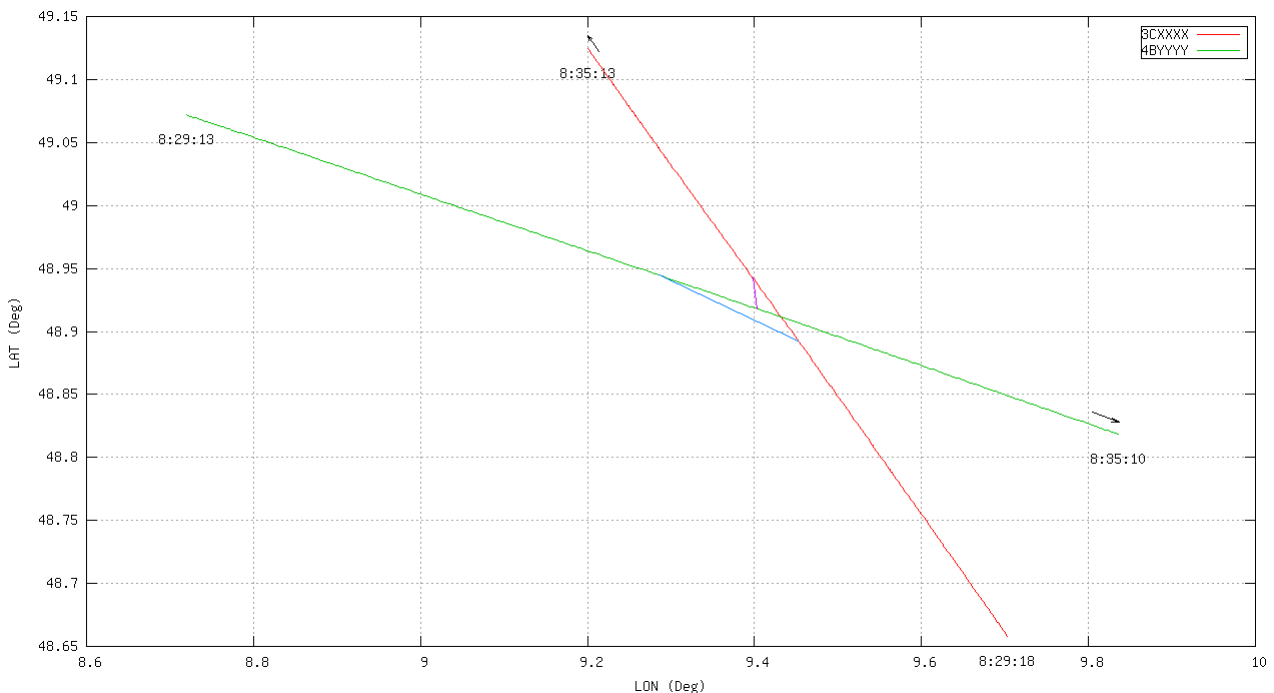


Bild 73: Horizontale Spuren ACAS-Event Beispiel 9

Bild 73 zeigt die horizontalen Bahnen beider Flugzeuge, in diesem Event finden keine horizontalen Manöver statt. Außerdem wird beim Vergleich von Bild 73 und Bild 72 (Höhendiagramm) klar, dass Grün noch vor der Kreuzung durch die Flugfläche von Rot hindurch stieg.

9.11 ACAS-Event Beispiel 10 – Sense Reversal

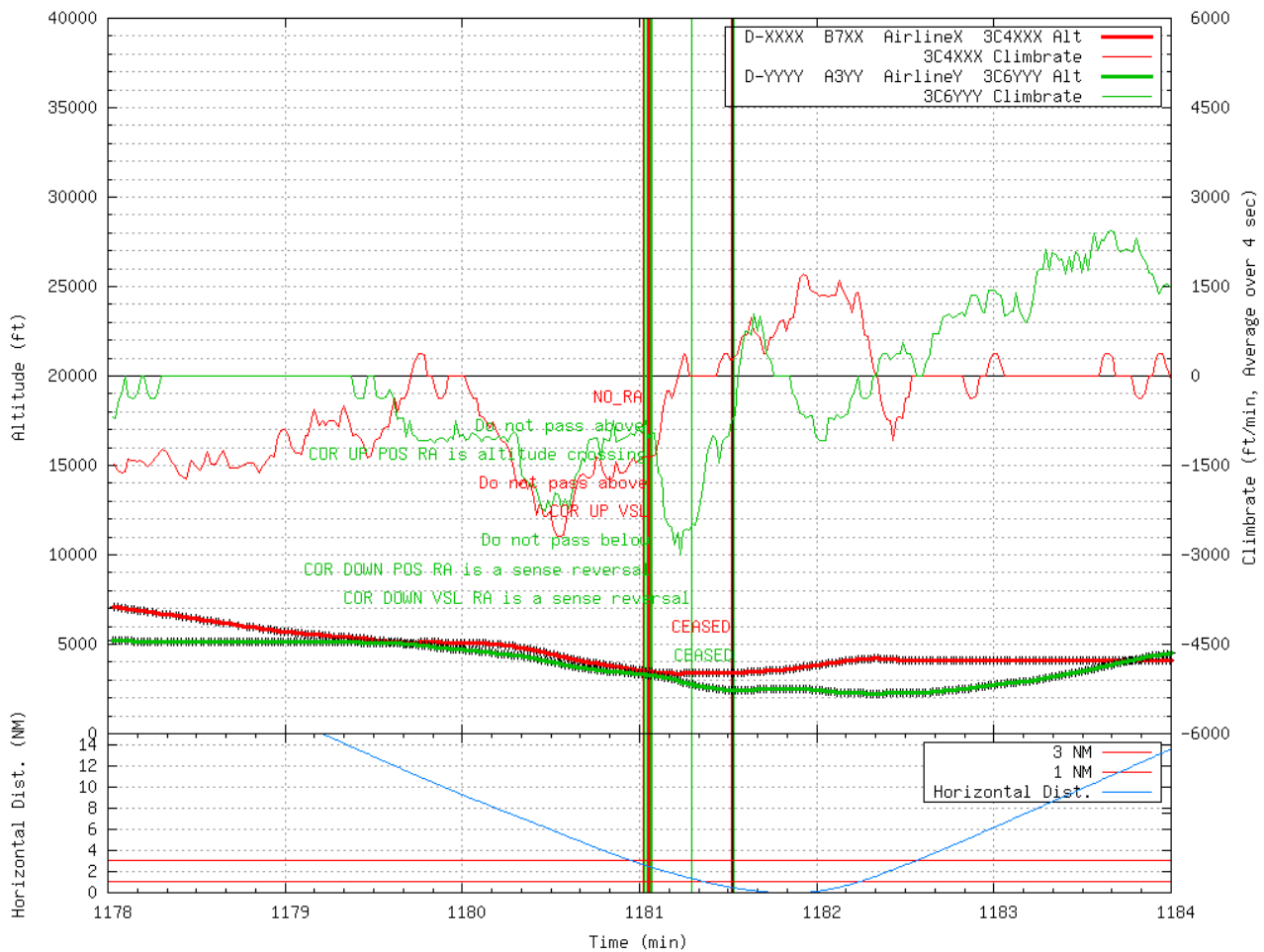


Bild 74: Höhendigramm ACAS-Event Beispiel 10 – Sense Reversal

ACAS kontrolliert sekundlich, ob das angewiesene RA noch der Situation entsprechend ist oder ein anderes RA angewiesen werden muss:

- Weil sich die Situation entspannt hat (Event Beispiel 2).
- Weil sich die Situation verschärft hat (Event Beispiel 14).
- Oder weil die ACAS-Geräte inkompatible RAs gewählt haben (dieses Beispiel).

Dabei kann ACAS das RA abschwächen, verstärken oder ganz umkehren. Das Umkehren des RA wird als Sense Reversal bezeichnet und wird dem Piloten und der Bedrohung angezeigt.

Bild 74 zeigt ein ACAS-Event, bei dem ein Sense Reversal stattfand. Grün spricht zuerst an und gibt ein Corrective **Upward** Positive Altitude Crossing RA (dt: „Steigen, kreuze die Höhe von Rot“). Rot hat zu Beginn kein RA (NO_RA), weist aber Augenblicke später selbst ein Corrective **Upward** Vertical Speed Limit RA (dt: „Beende Sinkflug“) an. Die beiden nach oben gerichteten RAs sind inkompatibel. Bei inkompatiblen RAs hat als letzten Schritt zur Lösung das ACAS-Gerät des Flugzeugs mit der kleineren Mode S-Adresse Vorrang. So setzt sich Rot durch und Grün wählt ein kompatibles RA Corrective **Downward** Positive RA und kennzeichnet es als Sense Reversal.

Zu Beginn des Events sind die Flugzeuge noch 2.6 NM horizontal und 200 ft vertikal separiert. Bei 1181 min 22 s sinkt der horizontale Abstand unter 1 NM, der vertikale beträgt dann 800 ft. Bei 1181 min und 46 s ist der horizontale Abstand 0,008 NM und der vertikale gerade wieder 1000 ft.

Bei diesem Event wird auch horizontal manövriert. Bild 75 zeigt die horizontalen Spuren, 3 min vor und nach dem Event, in einer Karte, auf der auch der Flughafen Frankfurt dargestellt ist. Der Beginn bzw. das Ende der RAs sind durch eine blaue bzw. eine violette Verbindungslinie markiert.

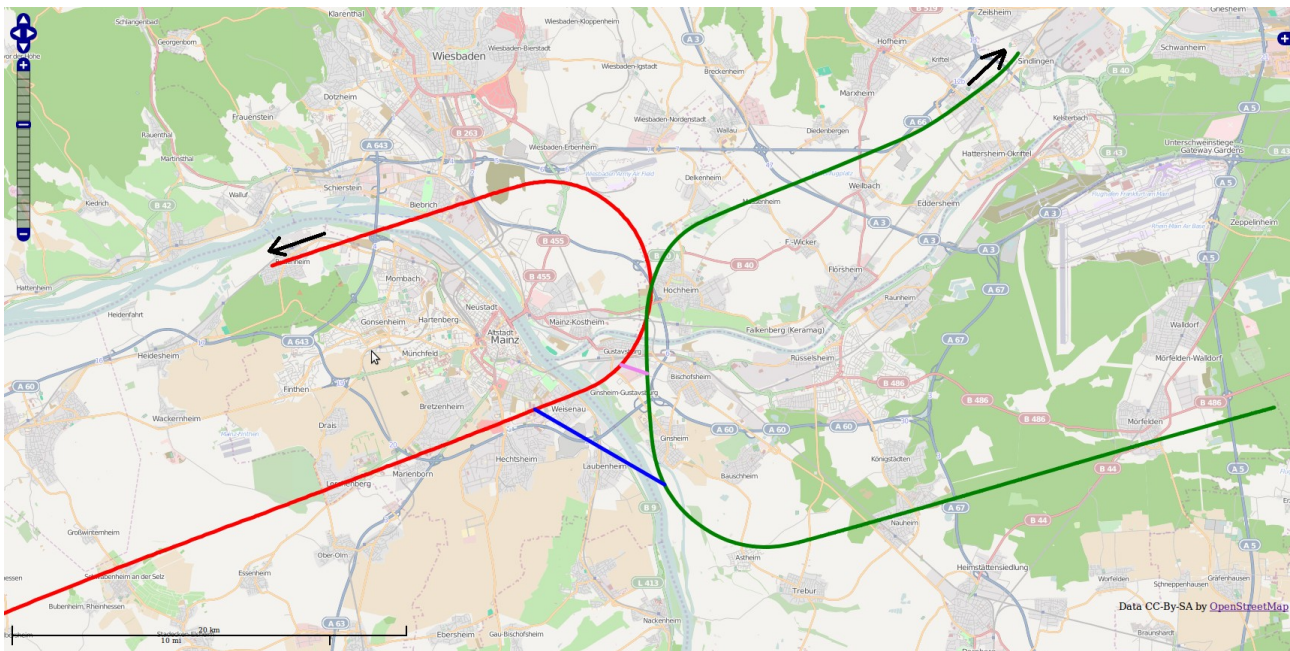


Bild 75: Horizontale Spuren ACAS-Event Beispiel 10 mit Flughafen Frankfurt

Nach geltenden Regeln würde dieser Vorfall als „schwere Störung“ klassifiziert werden. Er muss somit aufgearbeitet werden, um Lehren für die Weiterentwicklung und den Betrieb von ACAS zu ziehen. In diesem Event wird aber sehr deutlich, dass ein Pilot, der die ACAS-Anweisungen unter Zeitdruck umsetzen muss, in seinem nachträglichen Bericht den Vorgang nicht vollständig und zeitgenau schildern kann. Nur das ACAS-Monitoring mittels mithörenden Bodenstationen liefert eine zeitgenaue Dokumentation, ohne die die Lotsen- und Pilotenberichte nicht vollständig sind.

9.12 ACAS-Event Beispiel 11 – Verdrängung aus dem Reiseflug

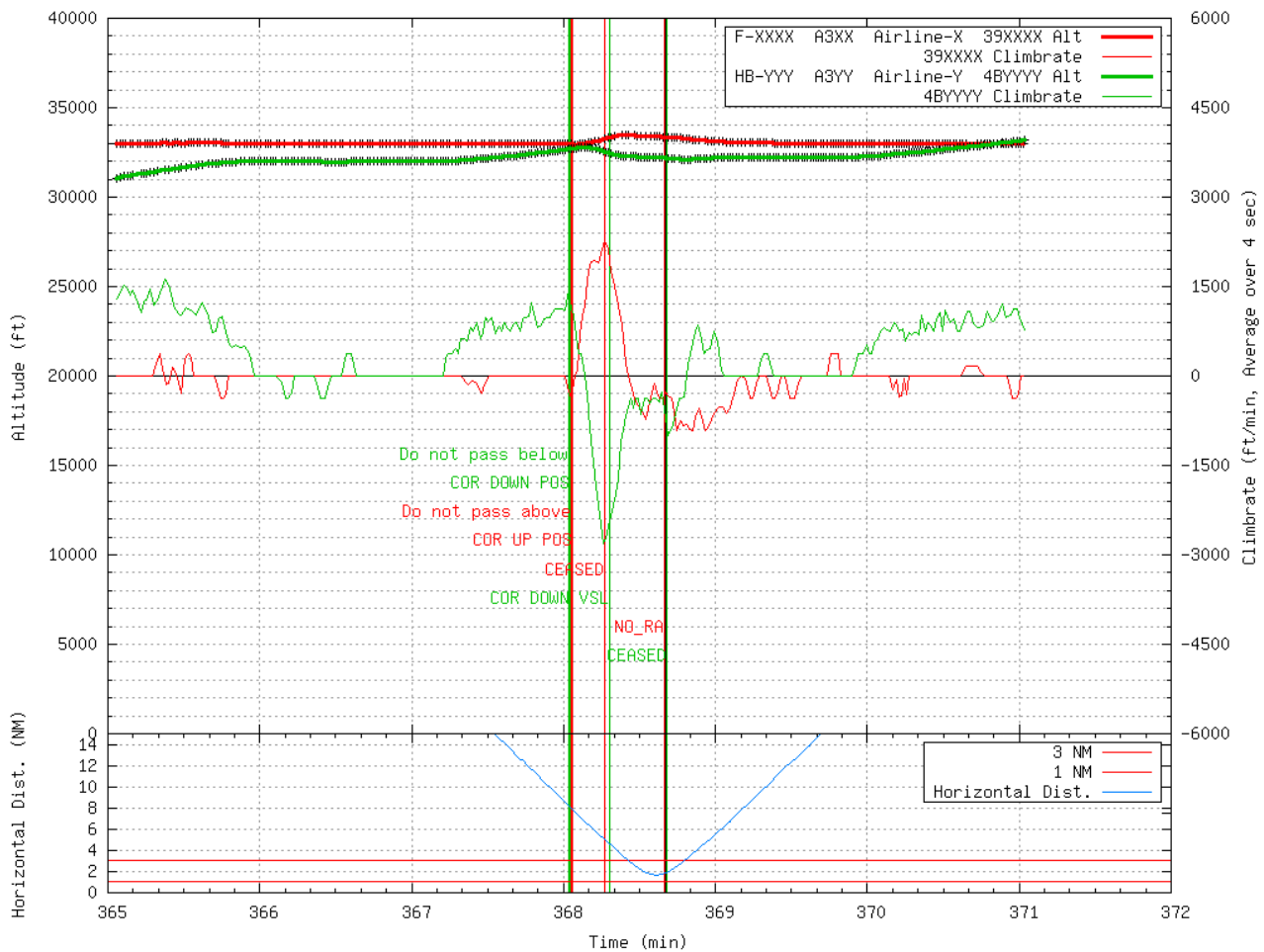


Bild 76: Höhendigramm ACAS-Event Beispiel 11 – Verdrängung aus dem Reiseflug

Bild 76 und Bild 77 zeigen eine Begegnung zweier Flugzeuge, bei der Rot aus dem Reiseflug verdrängt wird. Grün befindet sich 1000 ft unter Rot. Circa eine Minute vor dem horizontalen Passieren beginnt Grün zu steigen. ACAS erkennt einen Konflikt und löst in beiden Flugzeugen aus. Grün bekommt ein Corrective Downward Positive RA (dt: „Sinken!“), Rot ein Corrective Upward Positive RA (dt: „Steigen!“). Beide Flugzeuge reagieren umgehend. Bereits nach 15 s ist das RA in Rot beendet, das RA in Grün wird abgeschwächt. Nach weiteren 20 s ist auch das RA in Grün beendet. Rot kehrt auf seine alte Reiseflughöhe zurück.

Wegen der Dichte der Annäherung mussten beide ACAS-Geräte ansprechen. Der Manövrierende (Grün) musste sein Manöver unterbrechen, aber auch der eigentlich Unbeteiligte (Rot) muss seinen Reiseflug kurz unterbrechen.

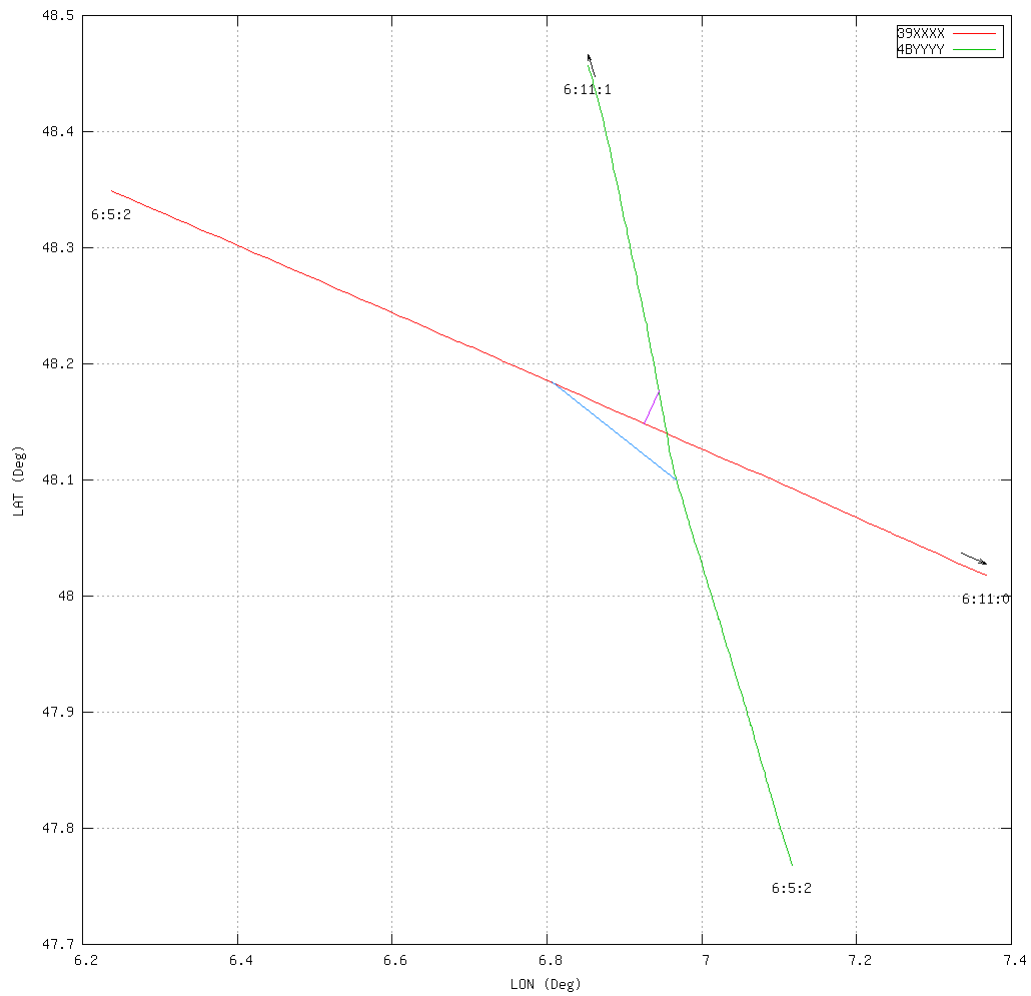


Bild 77: Horizontale Spuren ACAS-Event Beispiel 11

9.13 ACAS-Event Beispiel 12 – Langsame Reaktion

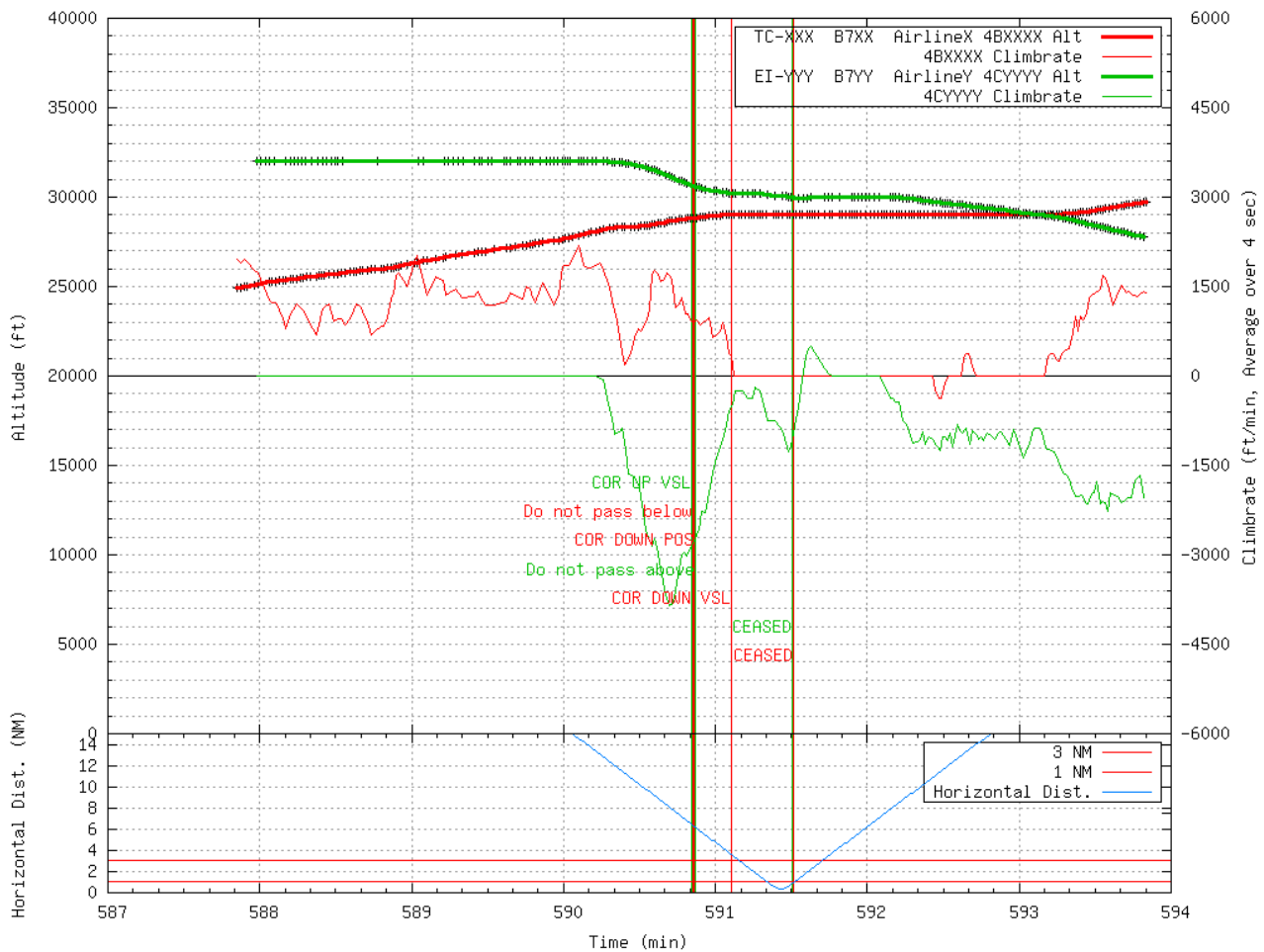


Bild 78: Höhendigramm ACAS-Event Beispiel 12 – Langsame Reaktion

Die Event-Beispiele 12 bis 16 zeigen verschiedene Reaktionen auf RAs. Das ACAS-Monitoring dokumentiert die Reaktionsbewegungen des Flugzeuges auf ein RA. Dabei wird ersichtlich, ob die Reaktion eines Flugzeuges den Anweisungen des ACAS gemäß Standard entspricht. ACAS-Event-Berichte eignen sich nicht, um etwaiges Fehlverhalten des Piloten nachzuweisen, sondern die Berichte sind vielmehr eine Quelle objektiver Daten zur vorurteilsfreien Suche nach Missständen im ACAS und in dessen Gebrauch.

Bild 78 und Bild 79 zeigen einen gewöhnlichen vertikalen Konflikt, in dem beide ACAS-Geräte ansprechen. Grün erhält ein Corrective Upward Vertical Speed Limit RA (dt: „Abfangen“). Grün hat sein Abfangen schon vor dem RA begonnen und kann es während des RA fortsetzen. Rot erhält das stärkere Corrective Downward Positive RA (dt: „Sinken!“). Während der ersten 15 s ist keine Reaktion von Rot zu beobachten, nicht einmal ein Abfangen. Die prompte Reaktion von Grün hat die Situation nach 15 s aber bereits entschärft und auch das ACAS in Rot kann sein RA zu einem VSL abschwächen. Da Rot exakt auf FL 290 abfährt, kann spekuliert werden, dass Rot entgegen dem Standard nicht die Anweisung des ACAS umsetzt, sondern die des Lotsen. Endgültige Klarheit kann nur das Pilotenprotokoll liefern, sofern es angefertigt wurde.

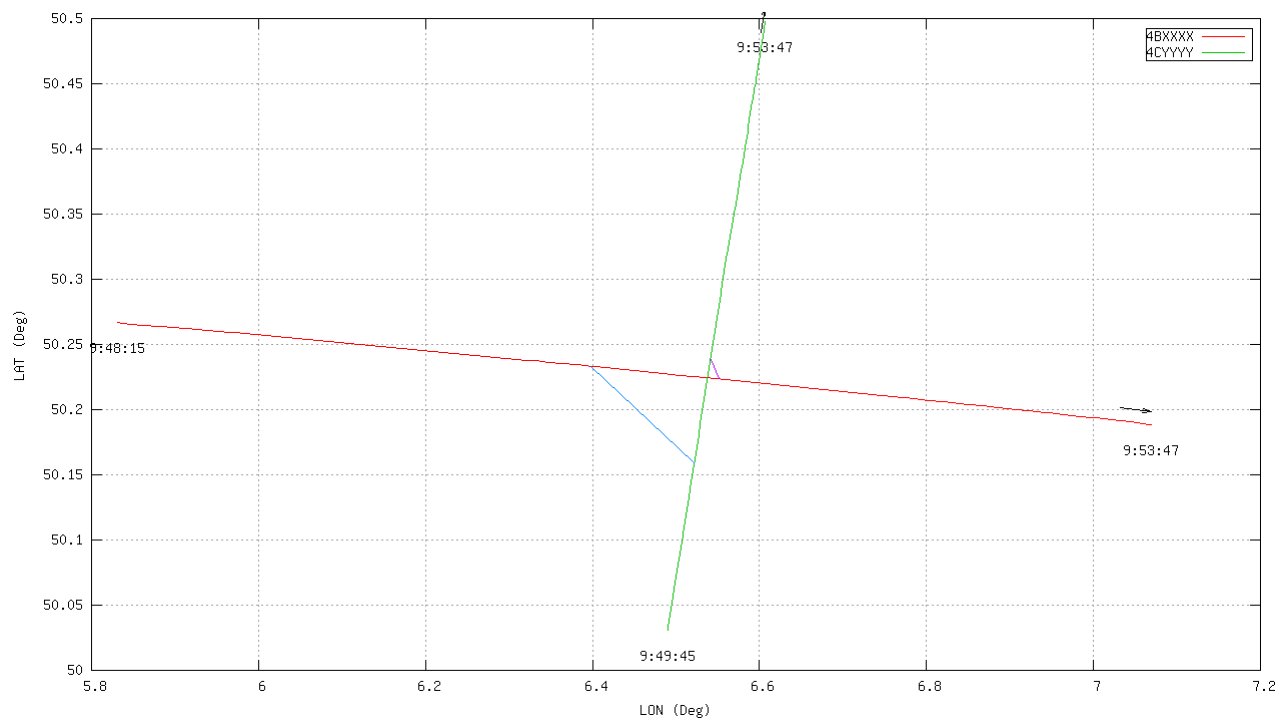


Bild 79: Horizontale Spuren ACAS-Event Beispiel 12

9.14 ACAS-Event Beispiel 13 – Keine Reaktion

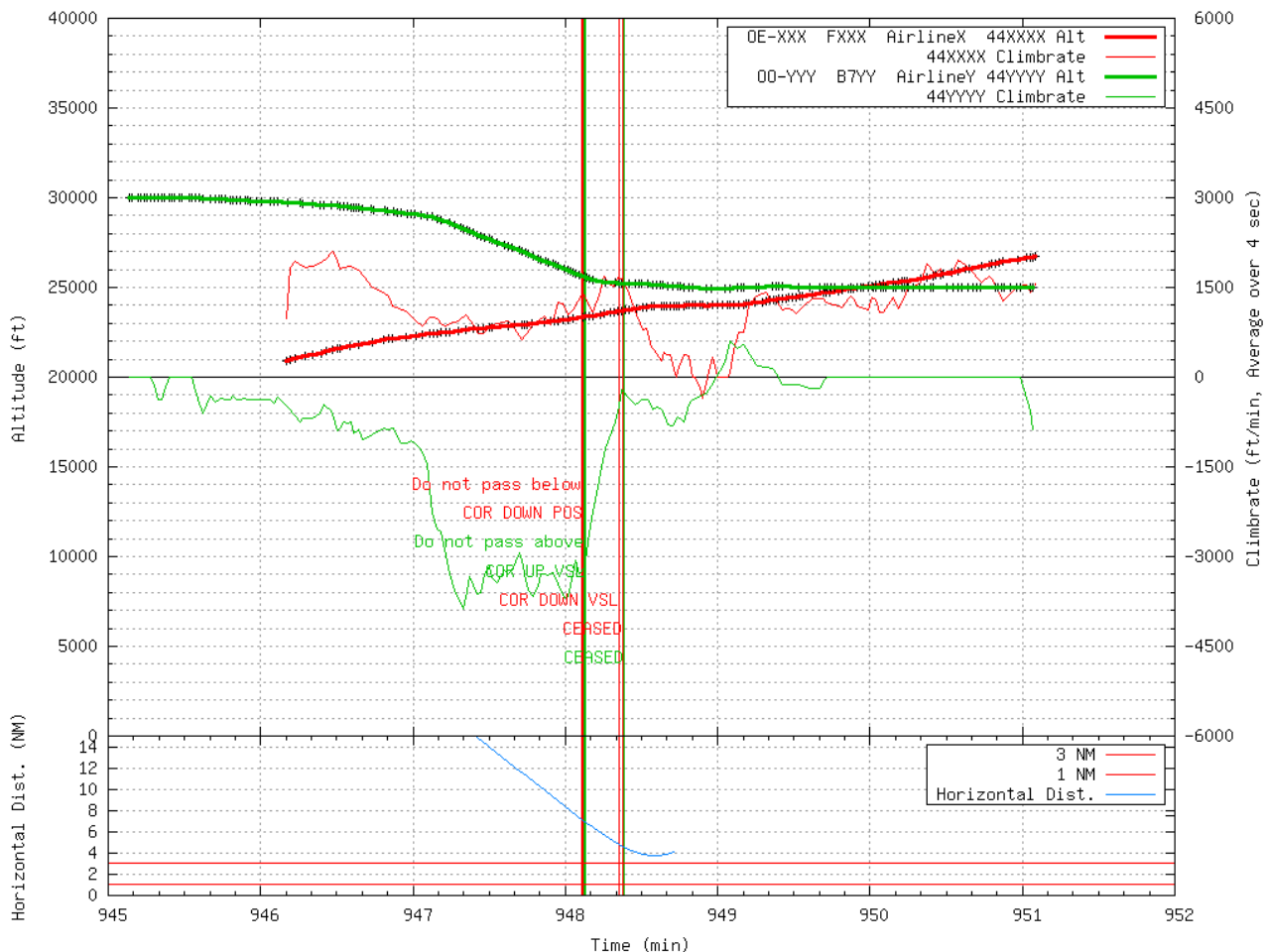


Bild 80: Höhendigramm ACAS-Event Beispiel 13 – Keine Reaktion

Dieser Beispiel-Event ähnelt stark dem vorhergegangenen. Bild 80 zeigt, dass Grün sein Corrective Upward Vertical Speed Limit RA (dt: „Abfangen“) umgehend umsetzt. Rot erhält ein Corrective Downward Positive RA (dt: „Sinken“), das während der 16 s Dauer des RA nicht einmal im Ansatz umgesetzt wird. Vermutlich weil Rot bei guten Sichtflugbedingungen sah, dass Grün bereits die projektierte Bahn von Rot gekreuzt hat (Bild 81). Rot fängt erst nach dem RA exakt auf FL 240 ab, wobei die Vermutung nahe liegt, dass es sich um die angewiesene Flughöhe handelt.

Dieses RA ist mit seiner Dauer von 16 s sehr kurz. Auch zeigt die horizontale Situation (Bild 81), dass sich die Flugzeuge schon passiert hatten, als ACAS ansprach. ACAS, das nur den Abstand, nicht die Richtung in den Test einfließen lässt, kann das horizontale Passieren jedoch erst feststellen, wenn der Abstand wieder zunimmt.

Sollte der Pilot aus dieser Überlegung bzw. entsprechender Erfahrung heraus die Reaktion auf das RA unterlassen haben, ist festzustellen, dass er die Situation hier richtig eingeschätzt hat. Die Gefahr ist nur, dass der Pilot das RA fälschlich einem nicht bedrohlichen dritten Flugzeug zuordnet und sich vor dem falschen Flugzeug in Sicherheit wähnen könnte. Diese Wahrscheinlichkeit wurde während der Standardisierung von ACAS in der ICAO als immer noch inakzeptabel hoch eingeschätzt und die standardkonforme Reaktion wäre, diesem RA zu folgen.

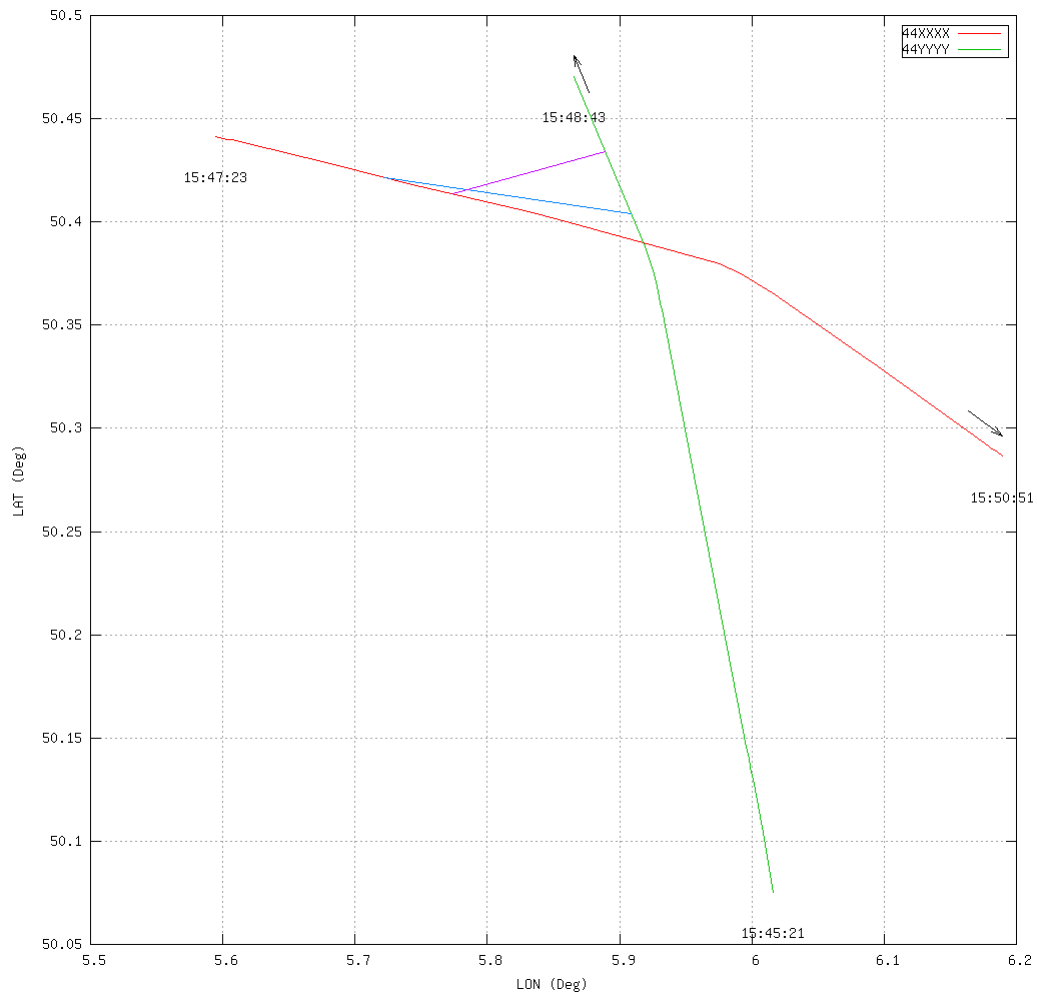


Bild 81: Horizontale Spuren ACAS-Event Beispiel 13

Der Event geschah am Rand der Reichweite des Netzes. Rot bewegt sich erst ca. 2 min vor dem RA in die Reichweite der Station. Die ADS-B-Spur von Grün reißt kurz nach dem RA ab und ab diesem Zeitpunkt kann der Abstand zwischen den Flugzeugen nicht mehr berechnet werden. In den vertikalen Spuren in Bild 80 sind daher auch 2 s bis 3 s große Lücken zwischen den Updates (schwarze Ticks).

9.15 ACAS-Event Beispiel 14 – Gegenteilige Reaktion

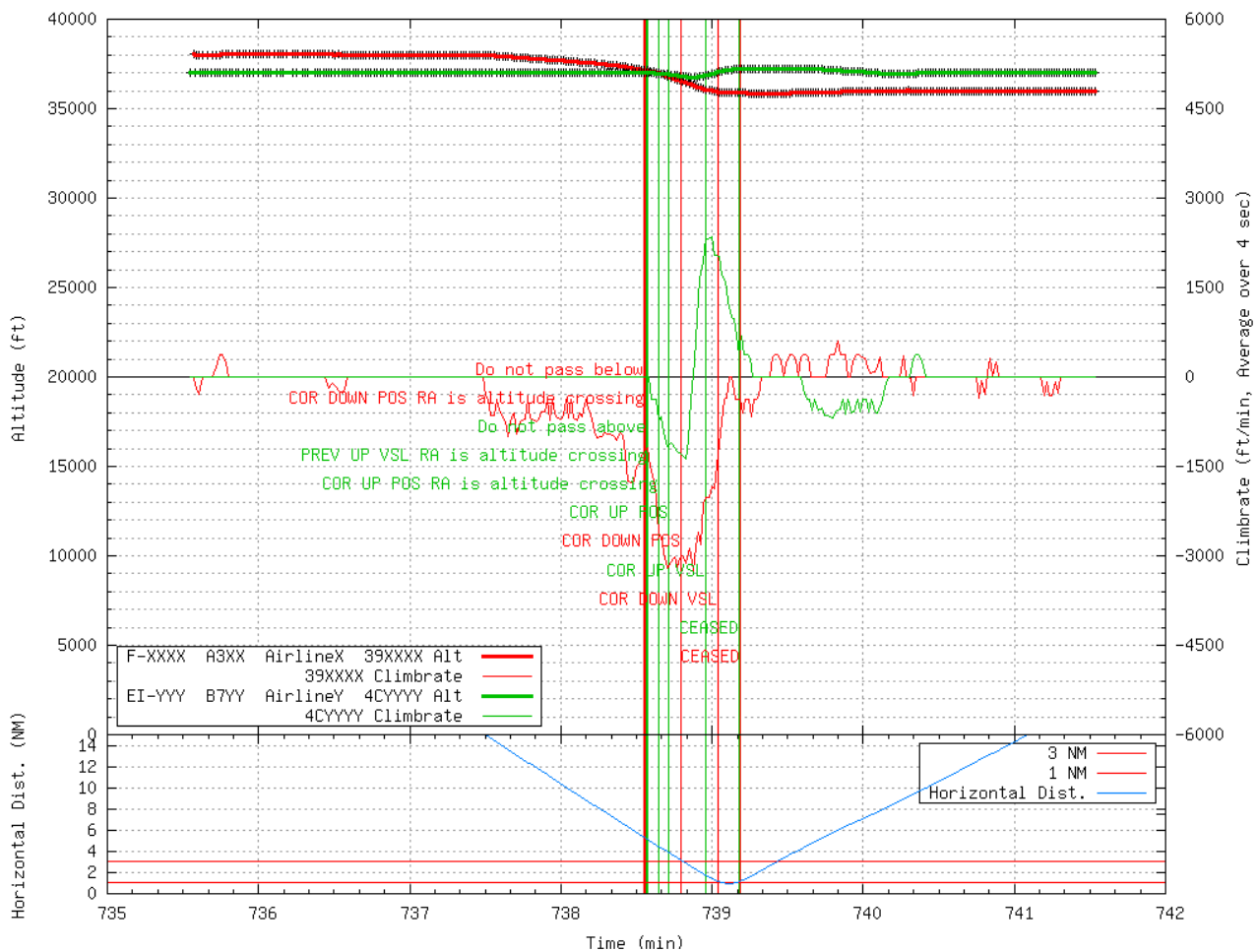


Bild 82: Höhendiagramm ACAS-Event Beispiel 14 – Gegenteilige Reaktion

Dies ist eines der komplexesten Events in dieser Arbeit. Es ereignete sich erst im Sommer 2009. Auch hier ist zur Orientierung zu beachten, dass die Beschriftungen direkt links an den entsprechenden senkrechten RA-Geraden angeordnet sind. Als ACAS in beiden Flugzeugen anspricht, sinkt Rot mit 1500 ft/min und ist noch 170 ft höher als Grün. Die beiden ACAS-Geräte müssen in dieser Situation Altitude Crossing RAs wählen, da Rot sowieso vor einem etwaigen Abfangen die Höhe von Grün passiert haben wird. Rot bekommt das Corrective Downward Positive Altitude Crossing RA, was zum Sinken mit 1500 ft/min auffordert (vgl. Tabelle 1 auf S. 21). Rot reagiert mit einer Verstärkung seiner Sinkrate auf 3000 ft/min.

Grün bekommt das **Preventive Upward** Vertical Speed Limit Altitude Crossing RA, das jedes Sinken verbietet und auf die bevorstehende Höhenkreuzung hinweist. Entgegen dem RA geht Grün in einen **Sinkflug** mit bis zu 1500 ft/min. 6 s später verstärkt das ACAS in Grün sein RA auf **Corrective Upward** Positive Altitude Crossing (dt: „Steigen“). Grün verstärkt sein **Sinken** weiter.

Inzwischen ist Rot niedriger als Grün und die ACAS-Geräte können das Altitude Crossing fallen lassen. Die Anweisungen „Sinken“ für Rot und „Steigen“ für Grün bleiben bestehen. Nach nun 20 s Sinkflug geht Grün in den geforderten Steigflug über. Als Grün nach weiteren 10 s die geforderte vertikale Rate erreicht hat, ist der Konflikt schon fast gelöst – beide ACAS-Geräte schwächen ihre RAs ab. Nach weiteren 10 s werden die RAs beendet. Trotz der gegenteiligen Reaktion von Grün konnte der Konflikt gelöst werden, weil Rot stärker reagiert hat, als angewiesen.

Offen bleibt, warum Grün das RA nicht befolgt hat. Das Preventive Altitude Crossing RA könnte missverstanden oder als fehlerhafte Anzeige gewertet worden sein. Oder Grün könnte Anweisungen vom Lotsen erhalten haben und diesen gefolgt sein, obwohl Piloten angewiesen sind, im Zweifel dem ACAS zu folgen [ACAS-Bulletin1]. Das Ausweichen entgegen der ACAS-Anweisung ist eine der Ursachen des Überlingen-Unglücks, das nur ca. 50 km entfernt geschah [BFU04].

Bei diesem Event wird auch ein Mangel des ACAS deutlich: Bei der Standardisierung konnte nur ein ACAS umgesetzt werden, das als „allerletzte Warnung“ fungiert. Würde ACAS weiter als 35 s vorausschauen dürfen, hätte es den Konflikt ohne Altitude Crossing lösen können. Allerdings würde ACAS bei einfacher Erhöhung der Vorwarnzeiten auch häufiger ansprechen.

Zur Erhöhung der Situation-Awareness sieht ACAS das Traffic Advisory (TA) vor, das bis zu 15 s vor dem RA gegeben wird. In den ACAS-Event Berichten sind keine TAs eingezeichnet, da das TA nur im Cockpit, aber in keiner ACAS-Nachricht angezeigt wird (vgl. Abschnitt 2.1). Das TA hat offensichtlich die Aufmerksamkeit von Grün geweckt – sonst hätte er nicht so prompt reagiert. Nur leider kann das TA nicht die eigentliche RA-Anweisung vorbereiten, da diese zum Zeitpunkt des TA noch nicht feststeht. Grün reagiert prompt aber falsch. Es ist Ziel des ACAS-Monitorings, solche Fälle zu dokumentieren, damit die nötigen Schlüsse für ACAS und seine Benutzung gezogen werden können.

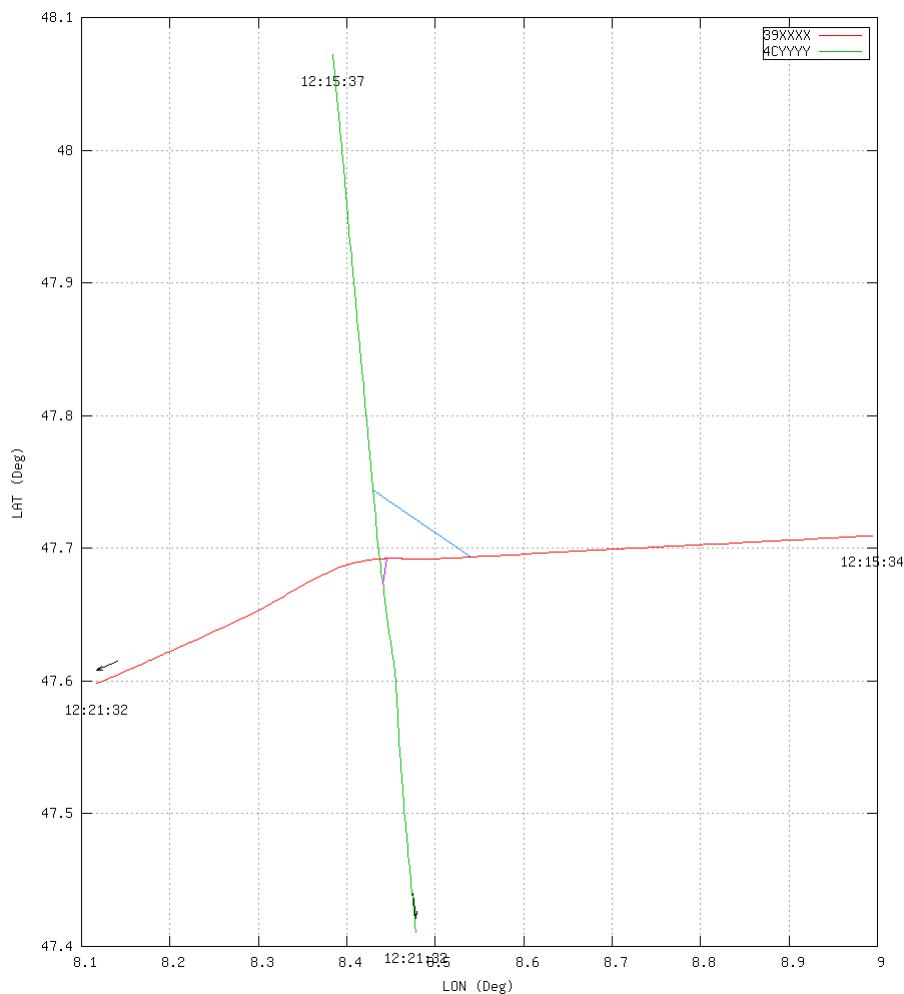


Bild 83: Horizontale Spuren ACAS-Event Beispiel 14

9.16 ACAS-Event Beispiel 15 – Horizontales Ausweichen, stärkere Reaktion als angewiesen



Bild 84: Höhendigramm ACAS-Event Beispiel 15 – Prompte deutliche Reaktion

Bild 84 zeigt einen normalen Konflikt. Beide ACAS-Geräte sprechen an und beide Flugzeuge reagieren prompt. Grün erhält die Anweisung zum Abfangen. Rot erhält die Anweisung mit 1500 ft/min (Corrective Upward Positive RA, vgl. Tabelle 1 in Abschnitt 2.7.5) zu steigen und geht aus einem Sinkflug mit 1500 ft/min innerhalb von 10 s kurzzeitig in einen Steigflug mit 4200 ft/min. ACAS hat selbst die Möglichkeit, einen Steigflug mit 2500 ft/min anzuweisen, dann wird das RA als Increased Rate gekennzeichnet, was hier aber nicht der Fall war.

Weiter ist in Bild 84 der „Knick“ im horizontalen Abstand auffällig. Der Grund ist in Bild 85 zu sehen – Grün führt ein horizontales Manöver, möglicherweise ein Ausweichmanöver, durch. Nach Sichtflugregeln hätte allerdings Rot ausweichen müssen. Da das Manöver noch während des RA eingeleitet wird, ist es nahe liegend, dass es sich um ein horizontales Ausweichen handelt. Ob dies tatsächlich so ist, kann nur ein eventuell vorhandenes Pilotenprotokoll klären.

In jedem Fall bleibt bemerkenswert, dass das ACAS-Monitoring auch die horizontale Bewegung detailliert dokumentiert und zusammen mit einem etwaig vorhandenen Pilotenprotokoll eine unzweifelhafte Beschreibung der Begegnung und der Gründe für den Ablauf liefert.

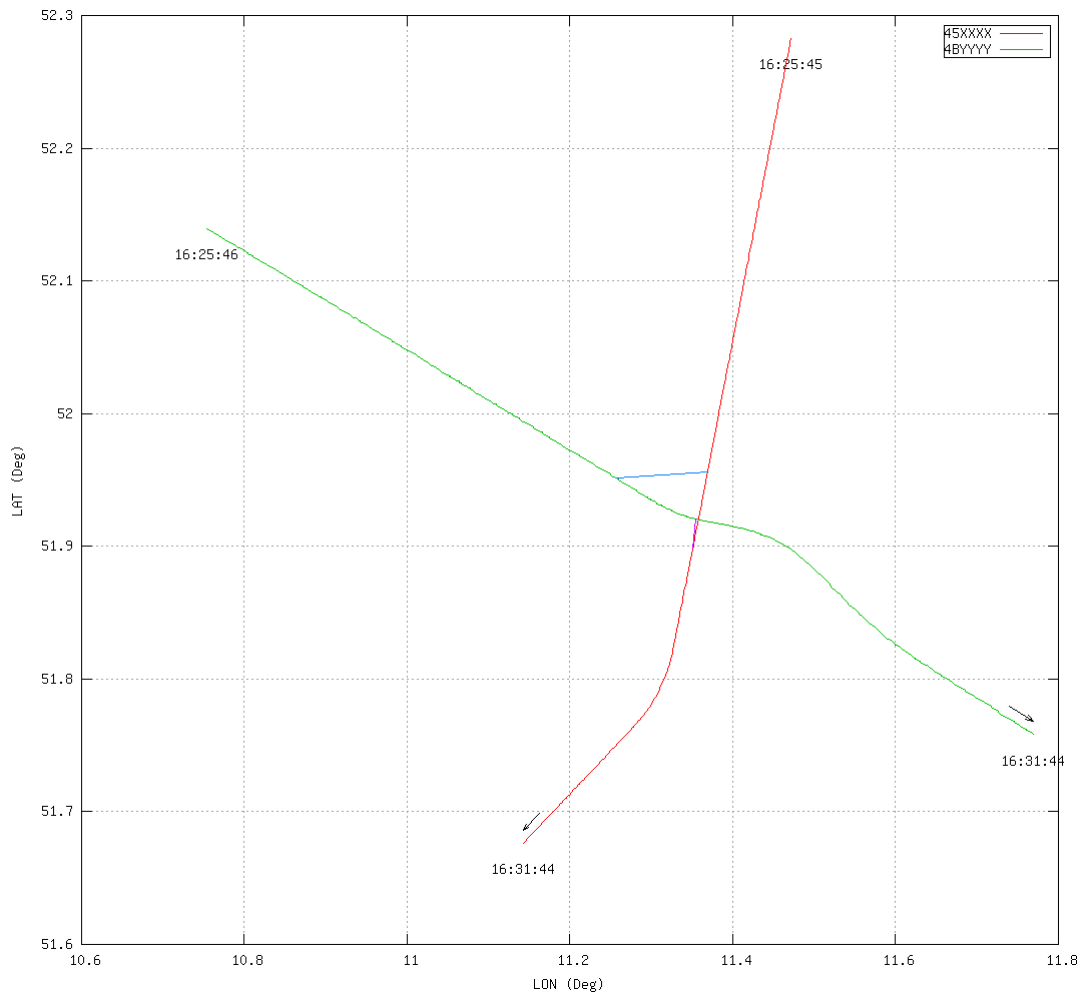


Bild 85: Horizontale Spuren ACAS-Event Beispiel 15 – Grün führt horizontales Manöver aus

Horizontale Manöver sind bei ACAS-Events im oberen Luftraum selten zu beobachten, denn ACAS gibt nur vertikale Anweisungen. Einen Vergleich von Sichtflugregeln und Ausweichanweisungen des ACAS bzw. TCAS gibt [Form86].

9.17 ACAS-Event Beispiel 16 – Überstarke Reaktion

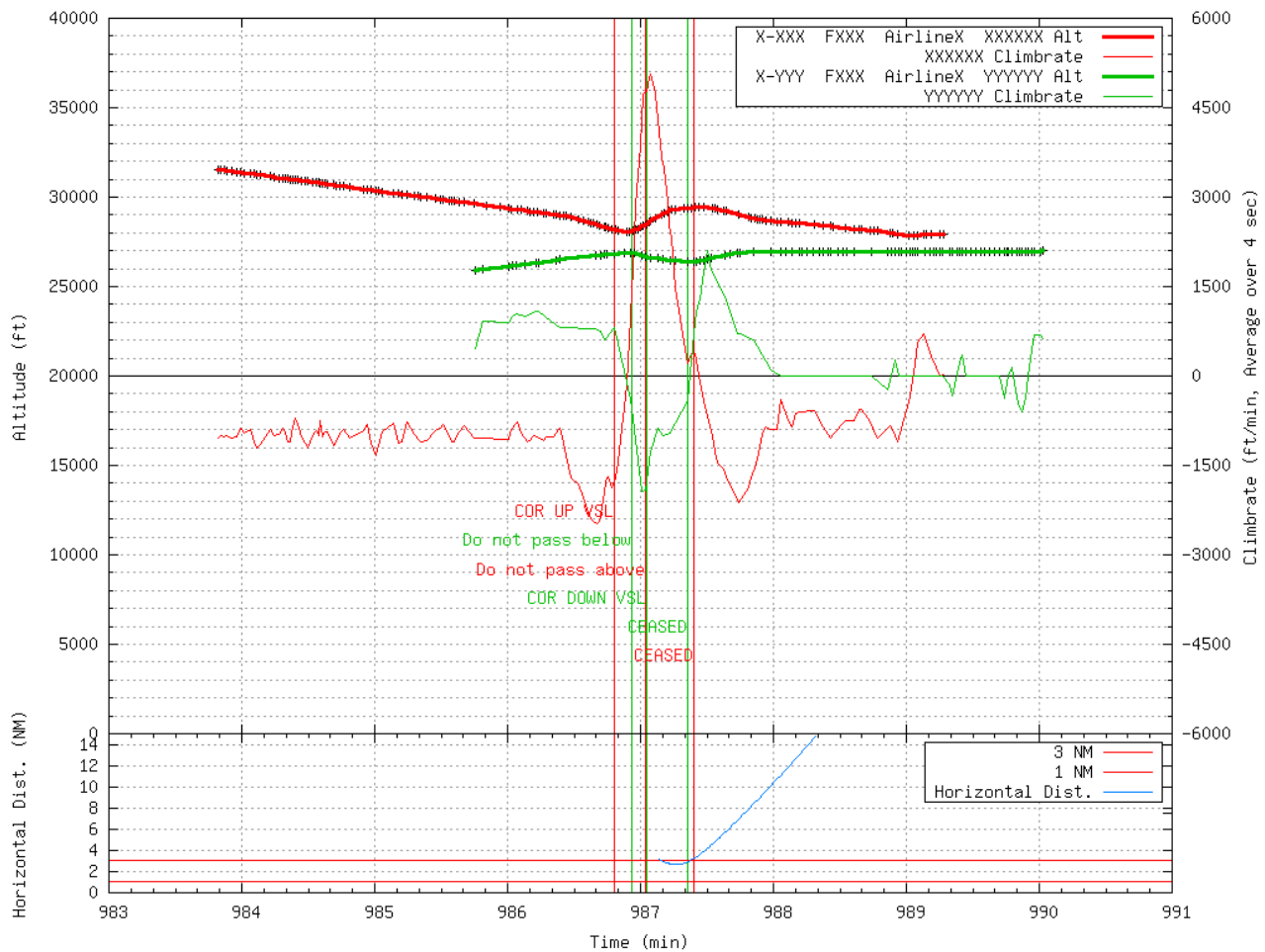


Bild 86: Höhendigramm ACAS-Event Beispiel 16 – Überstarke Reaktionen

Bild 86 und Bild 87 zeigen einen Event, bei dem übermäßig starke Reaktionen zu beobachten sind. Obwohl der Event am Rande des Empfangsbereiches (ca. 300 km von Antenne entfernt) stattfand, lassen sich auch hier noch Erkenntnisse über den Ablauf der Begegnung gewinnen. Vor dem RA sinkt Rot, Grün steigt. Rot erhält ein Corrective Upward Vertical Speed Limit RA und Grün ein Corrective Downward Vertical Speed Limit RA. Die RAs fordern zum Abfangen auf. Grün beendet nicht nur seinen Steigflug, sondern geht in einen Sinkflug mit 1500 ft/min über. Rot geht innerhalb von 13 s aus einem Sinkflug mit 1800 ft/min in einen Steigflug mit 5000 ft/min über. Damit reagieren beide Flugzeuge stärker als es ACAS angewiesen hat.

Die überstarke Reaktion löst zwar den aktuellen Konflikt, kann aber zu einem neuen Konflikt mit einem dritten Flugzeug führen. Ein interessantes Detail am Rande ist, dass beide Flugzeuge sehr ähnlich sind und von der gleichen Airline betrieben werden. Würde dieser Vorfall anhand der ACAS-Monitoring-Daten und der Pilotenberichte aufgearbeitet, könnten daraus wertvolle Anregungen für ein verbessertes ACAS-Training gezogen werden.

Bild 87 zeigt die automatisch erstellte Darstellung der horizontalen Spuren zu diesem Event. Grün kommt erst während des Events voll in den Empfangsbereich. Der Anfang des Events ist nur auf der roten Spur mit einem blauen Kreuz markiert. Das Ende des RA ist auf beiden Spuren mit einer violetten Verbindungslinie gekennzeichnet.

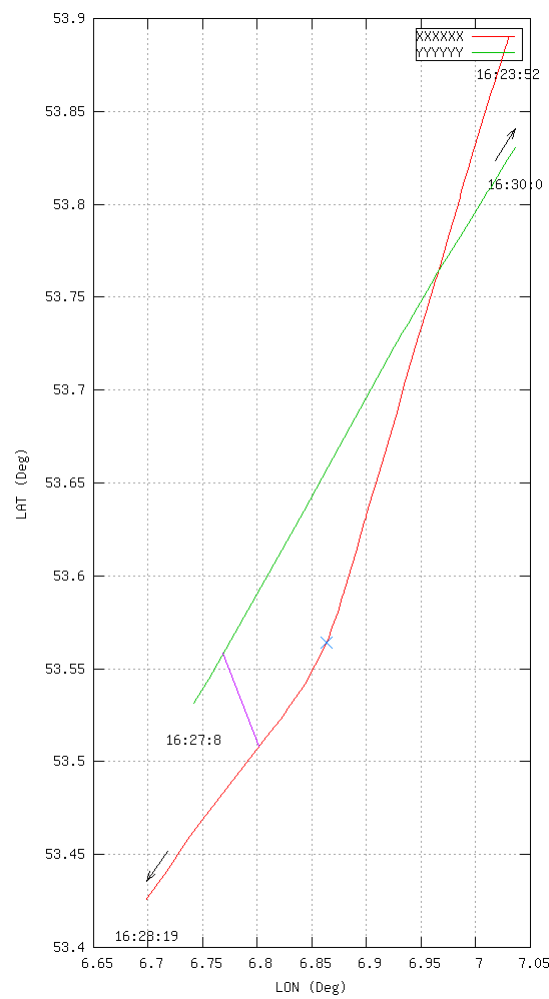


Bild 87: Horizontale Spuren ACAS-Event Beispiel 16

9.18 ACAS-Event Beispiel 17 – Nicht ACAS 7.0 Ausrüstung

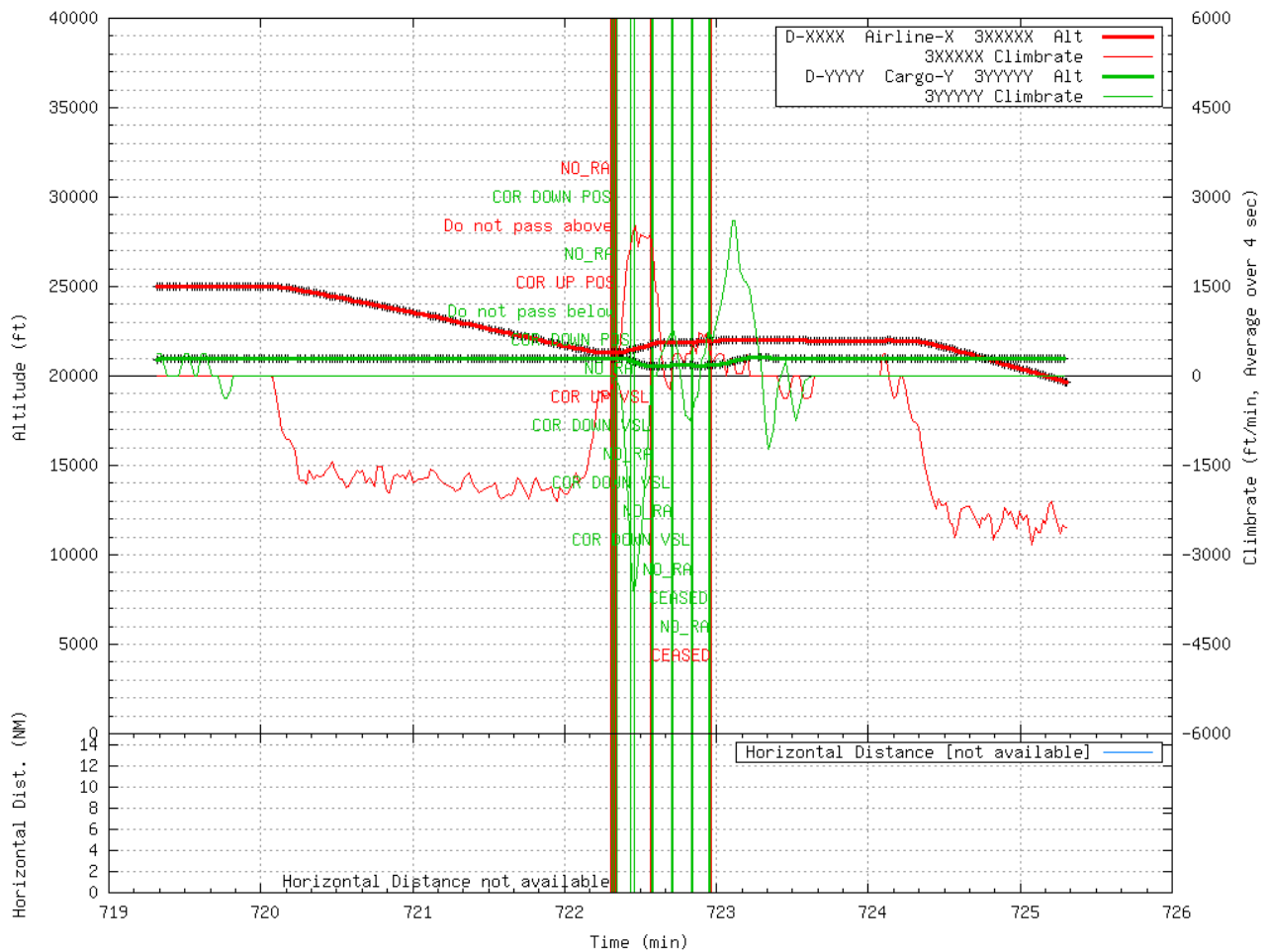


Bild 88: Höhendigramm ACAS-Event Beispiel 17 – Widersprüchliche Aussendungen von Grün

Bild 88 zeigt einen unübersichtlichen Verlauf des Events. Rot sinkt und verdrängt Grün aus dem Reiseflug. Auffällig an diesem Event ist die ACAS-Kommunikation: Grün meldet abwechselnd NO_RA und Corrective Downward Vertical Speed Limit.

Im Höhendigramm (Bild 88) ist die ACAS-Kommunikation nur vereinfacht dargestellt. Zur genauen Analyse der Kommunikation soll daher nun das Nachrichtendiagramm (Bild 89) betrachtet werden. Jeder Nachrichtentyp hat im Nachrichtendiagramm sein eigenes Symbol: UF16-30-Resolution-Message +, UF16-31-RA-Broadcast □ und DF16-30-Coordination-Reply ◇. Über der Zeit (Minuten seit Mitternacht) sind in der oberen Hälfte die Nachrichten von Rot dargestellt, in der unteren Hälfte die von Grün. In der obersten Zeile sind zu einem Flugzeug neben der Adresse **alle** Nachrichten zu ihren Empfangszeitpunkten eingetragen. In den Zeilen darunter sind die Inhalte der Nachrichten markiert.

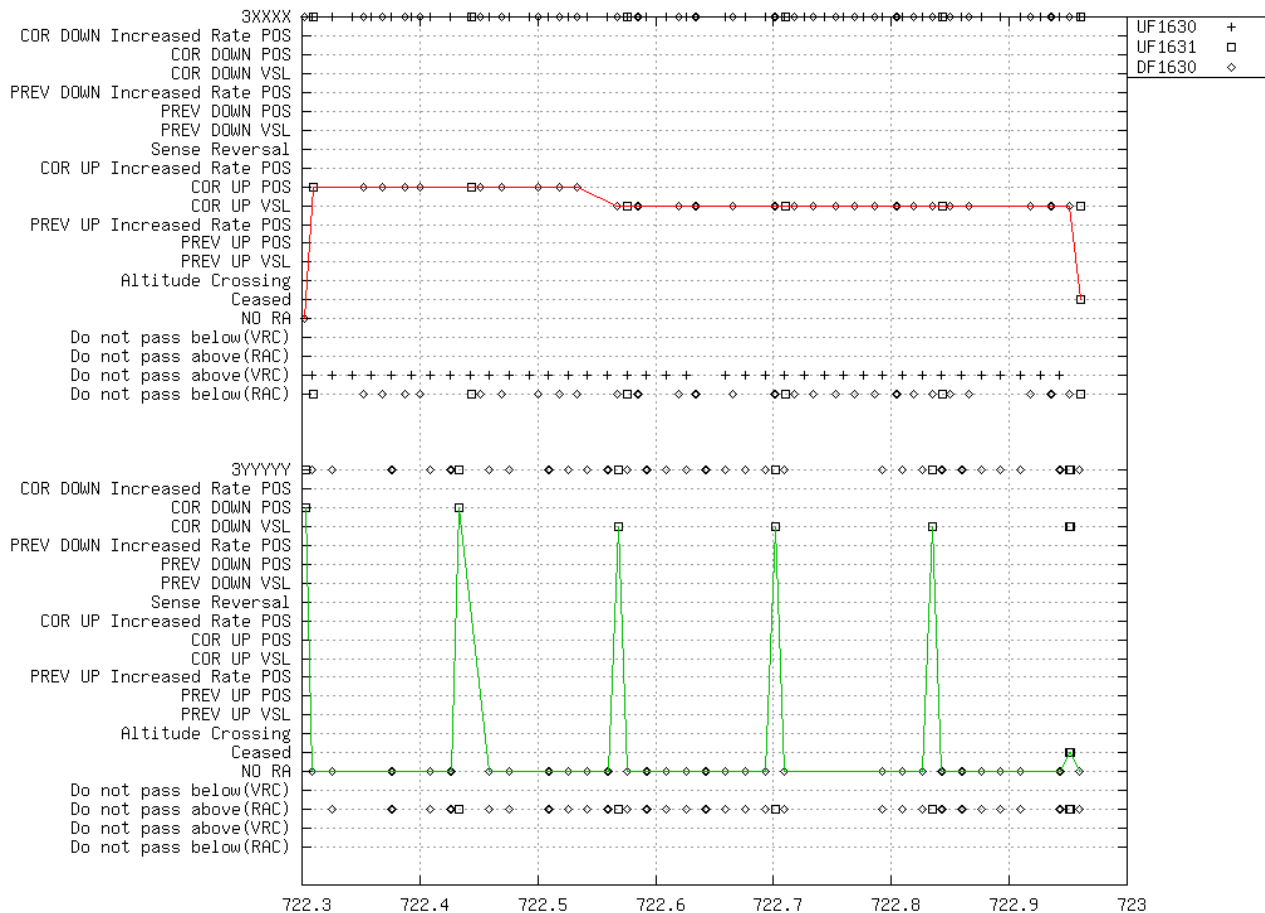


Bild 89: Nachrichtendiagramm – Widersprüchliche Aussendungen von Grün

Rot sendet erst eine DF16-30 \diamond mit NO_RA, dann eine UF16-30 + mit der Anweisung an Grün „Do not pass above“, dann eine UF16-31 \square mit dem eigenen Corrective Upward Positive RA und dem erhaltenen RAC „Do not pass below“. Nach zwei weiteren UF16-30 + mit der Anweisung an Grün „Do not pass above“, sendet Rot eine DF16-30 \diamond mit dem gleichen Inhalt wie in der UF16-31 \square (Corrective Upward Positive RA und der Quittierung des erhaltenen RAC „Do not pass below“), usw. Bei der Kommunikation ist später die Abschwächung des RA, aber kein Widerspruch zu erkennen.

Anders erscheinen die Nachrichten von Grün. Grün meldet zu Beginn des RA in einer UF16-31 \square ein Corrective Downward Positive RA. In der darauf folgenden DF16-30 \diamond jedoch NO_RA. Die nächste DF16-30 \diamond quittiert das von Rot zugewiesene RAC „Do not pass above“ und vermeldet wiederum NO RA. Diese DF16-30 \diamond wird mehrfach wiederholt. Erst die nächste UF16-31 \square meldet wieder das Corrective Downward Positive RA.

Es folgen sechs DF16-30 \diamond mit NO RA. Dann zeigt Grün mit einer UF16-31 \square die Abschwächung des RA zu Corrective Downward Vertical Speed Limit an. Die Inkonsistenz wird beibehalten, die UF16-31 \square melden ein RA, die DF16-30 \diamond melden NO RA.

Noch detaillierter ist die Kommunikation im Klartext dargestellt (Listing 11).

```

DF16-30 ◇ Coordination Reply
  Zeit: 722.32 min  Data: 80E18D9F30040100000000YYYYYY
  MTE: => No threat is being processed.
  ARA: => No vertical RA has been generated.
  RAC: => Do not pass above.

UF16-31 □ RA Broadcast
  Zeit: 722.43 min  Data: 8080000031E20100B3YYYYFFFFFF
  MTE: => One threat is being processed.
  ARA: => RA is corrective.
  ARA: => Downward sense RA has been generated.
  ARA: => RA is positive.
  RAC: => Do not pass above.

```

Listing 11: Vereinfachter Auszug aus dem Klartext – Exemplarisch eine UF16-31 und eine sich widersprechende DF16-30 von Grün

Im Klartext tritt der Widerspruch zu Tage: Nach [ICAO Annex10v4] sind die entsprechenden Bits (in Listing 11 fett) in beiden Nachrichten gleich codiert – die empfangenen Bits unterscheiden sich aber.

Die darauf folgenden Recherchen ergaben: Werden die Bits der DF16-30 ◇ nach einer veralteten Version des US-Standards [RTCA DO185] dekodiert, passen sie wieder zur UF16-31 □. Systeme nach diesem veralteten Standard werden im aktuellen US-Standard [RTCA DO185A] als TSO-C119A-Systeme [FAA TSO-C119A] bezeichnet, im weltweit gültigen Standard [ICAO Annex10v4] werden die veralteten Systeme gar nicht erwähnt.

Grün sendet also die DF16-30 ◇ nach veraltetem Standard, die UF16-31 □ nach aktuell gültigem Standard. Es ist zu vermuten, dass dies an einem veralteten Transponder liegt, denn das ACAS-Gerät sendet die UF16-31 □ selbst – die DF16-30 ◇ werden vom Transponder auf Anweisung des ACAS gesendet. Der Transponder ist also nicht zum gültigen Standard [ICAO Annex10v4] konform. Grün ist nicht vollständig mit ACAS 7.0 ausgerüstet. Ob diese teilweise ACAS 7.0-Ausrüstung dem ACAS-Mandat [AIC IFR 13/2003] entspricht und ob mit der veralteten Nachricht eine korrekte Koordination der ACAS-Geräte möglich ist, sprengt den Rahmen dieser Arbeit.

Der Betreiber von Grün wurde vom Institut kontaktiert und hat die Ergebnisse mit großem Interesse aufgenommen. Der Transponder ist im Rahmen einer anstehenden Überholung 2009 ausgetauscht worden. Weitere Untersuchungen ergaben, dass drei Prozent aller ca. 12.000 verschiedenen Flugzeuge, die sich zwischen Mai und Dezember 2008 im Empfangsbereich des wachsenden Netzes befanden, einen solchen veralteten Transponder besitzen.

Werden die falsch codierten Nachrichten ignoriert, so ergibt sich das „aufgeräumte“ Höhendia-gramm Bild 90.

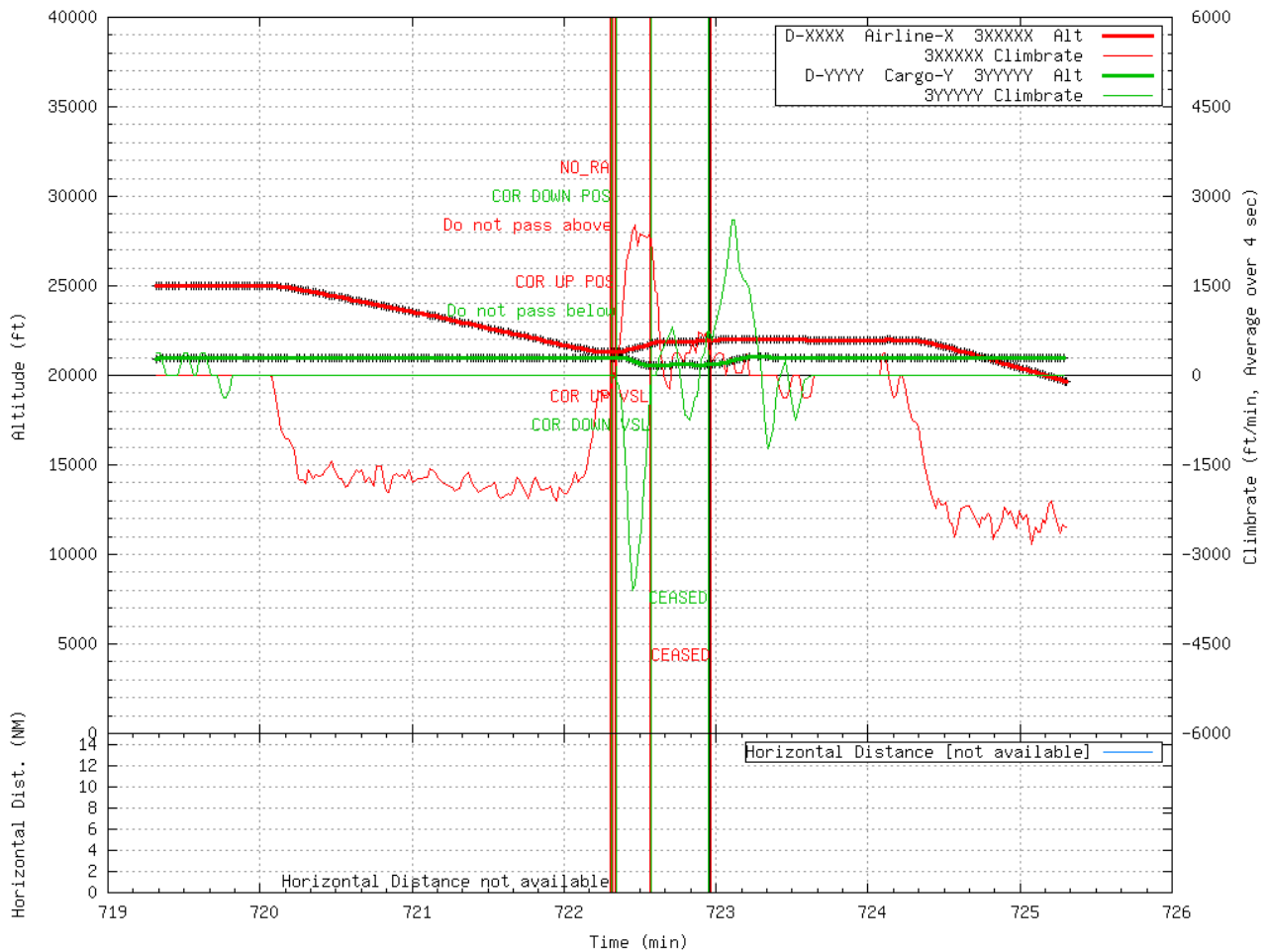


Bild 90: Höhendigramm – Widersprüchliche Aussendungen von Grün ignoriert

Dieser Event zeigt, dass ACAS-Monitoring nicht nur ACAS-Eingriffe dokumentiert, sondern auch Unstimmigkeiten in technischen Systemen aufzeigt und bei deren Analyse helfen kann.

Grün war nicht mit ADS-B ausgerüstet, weshalb kein horizontaler Abstand berechnet werden konnte.

9.19 ACAS-Event Beispiel 18 – Beabsichtigte Annäherung – Luftbetankung

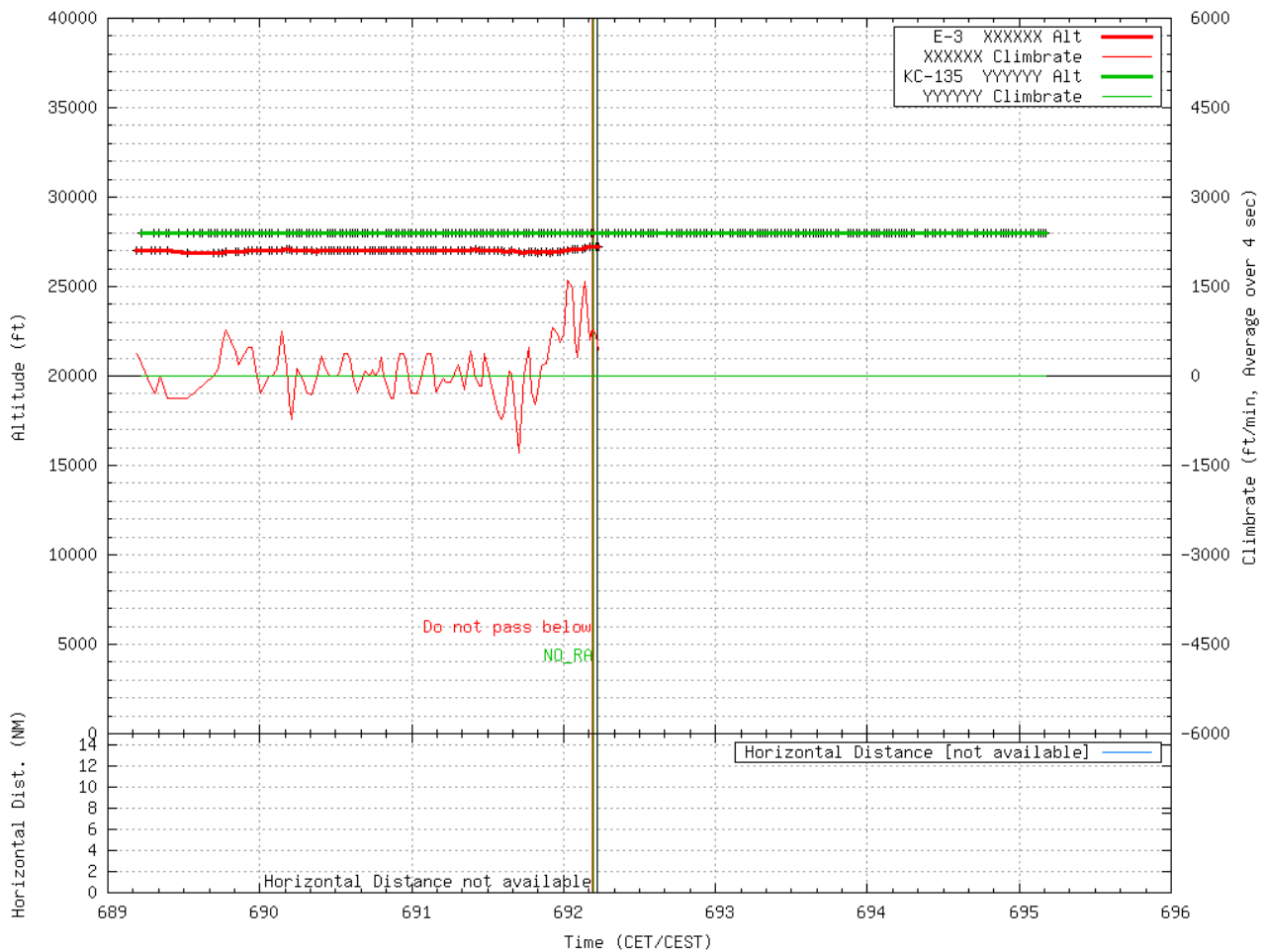


Bild 91: Höhendigramm ACAS-Event Beispiel 18 – Beabsichtigte Annäherung – Betankung

Im Allgemeinen kann man davon ausgehen, dass Annäherungen zwischen Flugzeugen nicht beabsichtigt sind. Bild 91 zeigt eine beabsichtigte Annäherung beim Betanken eines AWACS (Rot) durch eine KC-135 (Grün). Kurz vor dem Betanken spricht ACAS wegen der Annäherung an. Während des Betankens ist der Transponder des Tankenden (Rot) deaktiviert.

Die Nachbearbeitungsprogramme des ACAS-Monitors sondern beabsichtigte Annäherungen anhand der Adressen der beteiligten Flugzeuge automatisch aus. Das sind zum einen Events an denen **ausschließlich** militärische Flugzeuge beteiligt sind, dazu zählen auch die zivilen Trainings-Flugzeuge der Bundeswehr (GFD). Zum anderen sind das wissenschaftliche Begegnungen unter kontrollierten Bedingungen zwischen den Flugzeugen des DLR, z. B. zur Erprobung von Onboard-Radar-Systemen.

Beabsichtigte Events gehen in keine Statistik ein. Keines der hier gezeigten Events (außer Nr. 18) ist eine derartige beabsichtigte Begegnung.

9.20 Statistik der automatischen Dauerauswertung

Seit dem 20. April 2007 wird eine kontinuierliche Dauerauswertung von Mode S-Daten durchgeführt. Das erste Jahr mit der experimentellen Station auf dem Dach des Institutes in Braunschweig, seit dem 30. April 2008 mit der Doppelstation am DLR-Braunschweig und seit August 2008 mit allen fünf Stationen des ACAS-Monitor-Netzes (Kapitel 8). Die vollständig automatisierte Aufzeichnung und Auswertung dauert bis heute (20. September 2010) an. Dieser Abschnitt beschreibt die Statistik mit Stand 31. August 2009.

Die experimentelle Station in Braunschweig auf dem Institutsdach des IfEV beobachtete tagsüber (07:00-21:00 Uhr MEZ) ca. 300 ACAS-ausgerüstete Flugzeuge gleichzeitig im Empfangsbereich (vgl. Bild 12 S. 36). Die Braunschweiger Station auf dem DLR-Turm beobachtete tagsüber ca. 450 ACAS-Flugzeuge gleichzeitig. Das gesamte Netz beobachtete tagsüber ca. 700 ACAS-Flugzeuge gleichzeitig (vgl. Bild 13 S. 37). Wegen technischer Defekte und wegen eines Umzugs war die Station in Oberpfaffenhofen seit November 2008 nur teilweise und seit August 2009 vollständig offline. Ohne Oberpfaffenhofen wurden tagsüber vom restlichen Netz nur ca. 600 Flugzeuge beobachtet. Damit lässt sich die Gesamtzahl der beobachteten Flugstunden abschätzen:

Es wurden vom 20. April 2007 bis 31. August 2009 ca. 5.000.000 Flugstunden von ACAS-ausgerüsteten Flugzeugen beobachtet.

In dieser Zeit wurden ca. 2,5 TByte komprimierte Mode S-Daten aufgezeichnet, ausgewertet und archiviert. Ein Tag, an dem alle Stationen vollständig in Betrieb sind, erzeugt ca. 4 GByte komprimierte Mode S-Daten, entpackt sind das ca. 50 GByte Daten, die ausgewertet werden müssen.

Insgesamt wurden **1582** ACAS-Events beobachtet. Zusätzlich wurden noch 617 beabsichtigte Annäherungen (vgl. Abschnitt 9.19) beobachtet, die aber nicht gezählt und ausgewertet werden. Von diesen 1582 ACAS-Events konnte bei **703** Events nur ein beteiligtes Flugzeug ermittelt werden. Dies geschieht, wenn nur ein Flugzeug mit ACAS ausgerüstet ist oder wenn sich das Event am Rande des Empfangsbereichs ereignet und keine UF16-30-Resolution-Message empfangen wurde. Bei den übrigen **879** konnten beide Event-Partner ermittelt werden. Von diesen 879 vollständigen Events ist bei **669** Events ein Flugzeug mit ADS-B ausgerüstet, bei **340** Events sogar beide (vgl. Bild 92).

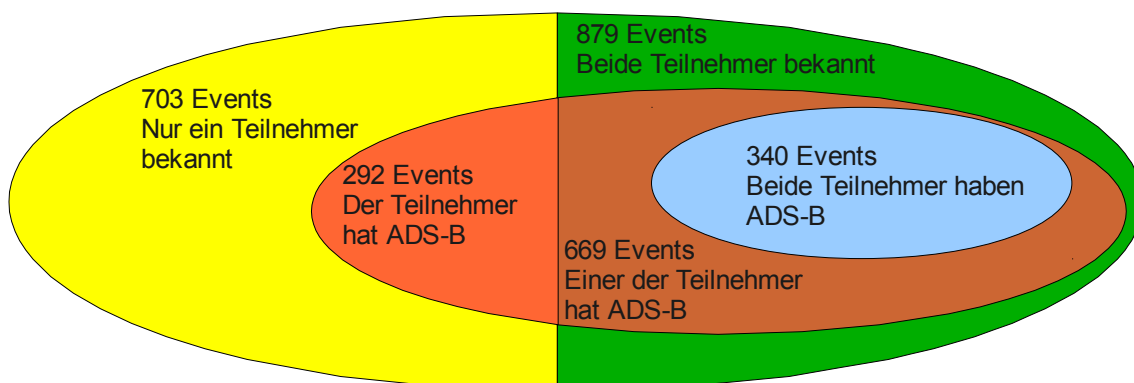


Bild 92: Anteil der vollständig beobachteten Events und der ADS-B-Ausrüstung

9.21 Orte der beobachteten ACAS-Events

Bei 961 beobachteten Events war mindestens eines der beteiligten Flugzeuge mit ADS-B ausgerüstet. All diese Events sind in der Karte in Bild 93 eingetragen.

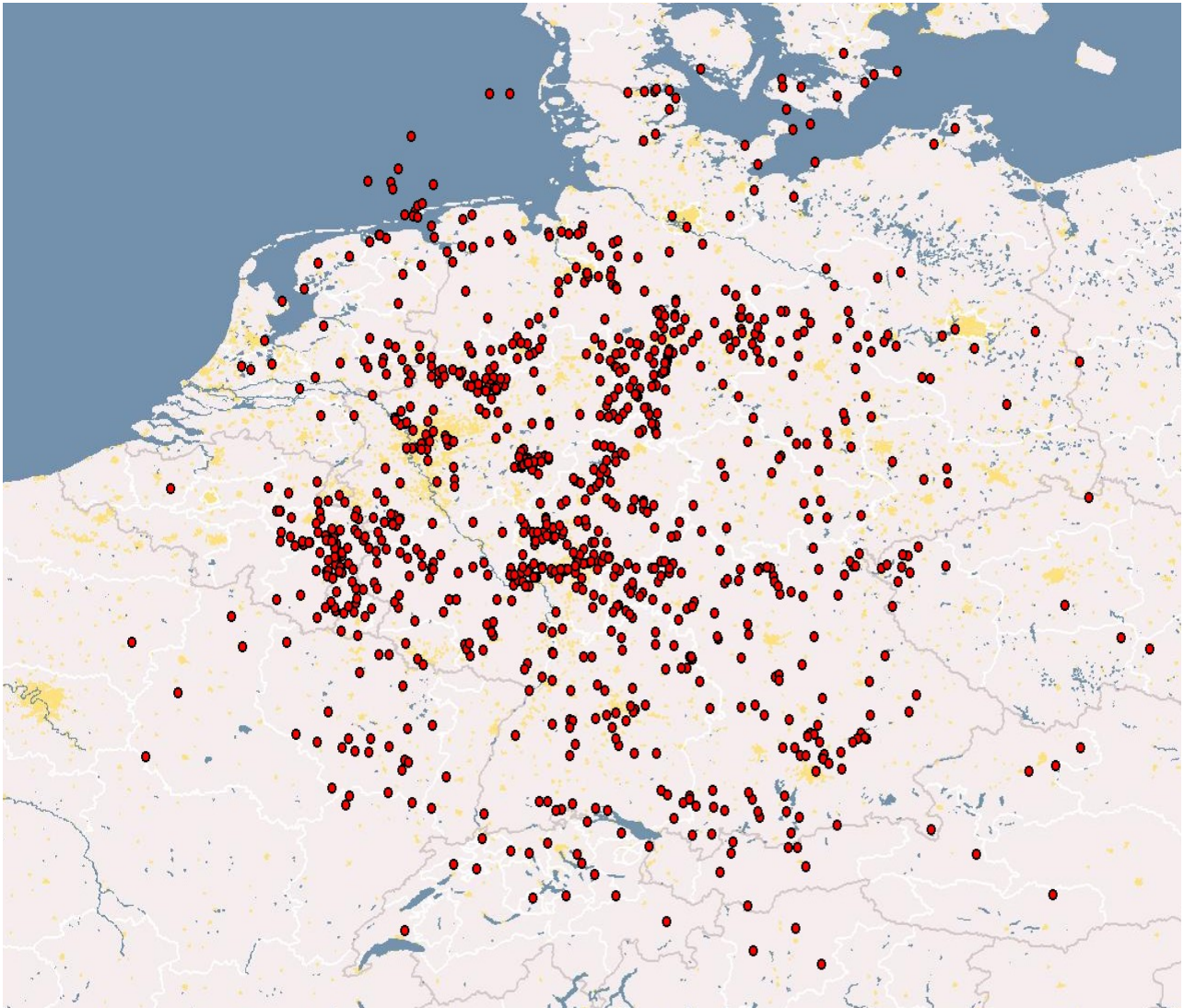


Bild 93: Orte der 961 lokalisierbaren ACAS-Events, beobachtet seit April 2007 in Norddeutschland und seit August 2008 mit den fünf Stationen des ACAS-Monitor-Netzes. Stand 31. August 2009

10 Zusammenfassung

Das weltweit standardisierte Kollisionsschutzsystem für Verkehrsflugzeuge ACAS (bzw. TCAS) erteilt dem Piloten eine Ausweichanweisung (Resolution Advisory, RA), wenn die Gefahr besteht, in den nächsten 15 s bis 35 s mit einem anderen Flugzeug zu kollidieren.

In dieser Arbeit wurde ein ACAS-Monitoring-System zur automatischen, zeitgenauen Dokumentation der Anweisungen des Kollisionsschutzsystems entwickelt, erprobt und beschrieben. Damit setzt diese Arbeit als erste die Forderung der ICAO nach Bodenstationen um, die ACAS-Meldungen aufzeichnen und analysieren. Die ICAO sah dazu im ACAS-Standard eine Meldung vom Flugzeug an Bodenstationen vor, den sog. UF16-31-RA-Broadcast. Der in dieser Arbeit entwickelte ACAS-Monitor geht einen Schritt weiter und wertet die komplette Kommunikation der ACAS-Geräte automatisch aus, um eine umfassende Rekonstruktion der ACAS-Eingriffe (RA) in den Luftverkehr zu erstellen. Die automatisch erstellten Berichte zeigen die Verkehrssituation, die zum Ansprechen des ACAS geführt hat, das eigentliche RA mit der ACAS-Kommunikation und die Reaktion der Flugzeuge.

ACAS-Geräte an Bord von Flugzeugen benutzen die Frequenzen (1030 MHz und 1090 MHz) und Protokolle des Mode S-Sekundärradars der Flugsicherung. Das ACAS-Gerät überwacht alle Flugzeuge im Empfangsbereich, indem es aktiv alle erreichbaren Flugzeuge abfragt. Durch mithörende Bodenstationen auf diesen beiden Frequenzen werden die Kommunikationen der Flugzeuge mit dem Sekundärradar, die ACAS-Kommunikationen der Flugzeuge untereinander und die ADS-B-Positionsmeldungen der Flugzeuge empfangen. Diese Aussendungen, insbesondere die ACAS-Kommunikationen, werden in dieser Arbeit zur Rekonstruktion der ACAS-Eingriffe herangezogen.

Den Kern des entwickelten Rekonstruktions-Algorithmus bildet ein Umgebungsmodell, das mithaltend anhand der Empfangsdaten aktualisiert wird. Das Umgebungsmodell speichert: Welche Flugzeuge sich im Empfangsbereich befinden. Welche Höhe jedes Flugzeug hat, mit ca. sekundlicher Erneuerung. Und welche Position ein Flugzeug hat, sofern es mit ADS-B ausgerüstet ist. Erkennt das ACAS-Gerät in einem Flugzeug eine drohende Kollision, so gibt es eine Anweisung an den Piloten und koordiniert sich mit dem anderen Flugzeug. Der ACAS-Monitor empfängt auch diese Koordinationsnachrichten. Daraus wird anhand des Umgebungsmodells der Koordinationsdialog rekonstruiert, und jede einzelne Nachricht und der Dialog im Zusammenhang werden auf Standardkonformität geprüft. Der Dialog wird mit den Spurdaten aus dem Umgebungsmodell zu einem ACAS-Event-Bericht zusammengestellt und ausgegeben.

Die für das ACAS-Monitoring nötigen Bodenstationen sind um einen 1030 MHz Empfänger erweiterte ADS-B-Bodenstationen, wie sie wegen der im Vergleich zu Radaranlagen günstigen Anschaffung weltweit mehr und mehr Verbreitung finden. Mit den Ergebnissen dieser Arbeit lassen sich diese erweiterten ADS-B-Bodenstationen auch zum ACAS-Monitoring nutzen. Damit lässt sich die Funktion des Kollisionsschutzsystems ACAS, des letzten technischen Sicherheitsnetzes vor einer Kollision, effektiv überwachen und dokumentieren. Die Analyse der ACAS-Event-Berichte kann Probleme im ACAS und dessen Benutzung aufzeigen, bevor diese Probleme im Verbund mit anderen etwaigen Defiziten zu Katastrophen führen könnten.

Literaturverzeichnis

URLs zuletzt geprüft am 30. November 2009

Literatur aus der Flugsicherung

- [ACAS-Bulletin1] EUROCONTROL: *ACAS II-Bulletin Nr. 1 – Follow the RA!* Brüssel 2005
Download: [://www.eurocontrol.int/msa/gallery/content/public/documents/ACAS_Bulletin%20_1_disclaimer.pdf](http://www.eurocontrol.int/msa/gallery/content/public/documents/ACAS_Bulletin%20_1_disclaimer.pdf)
- [ACAS-Bulletin6] EUROCONTROL: *ACAS II-Bulletin Nr. 6 – Incorrect Use of the TCAS Traffic Display* . Brüssel 2005
Download: http://www.eurocontrol.int/msa/gallery/content/public/documents/ACAS_Bulletin_6_disclaimer.pdf
- [AIC IFR 13/2003] DFS: *Aeronautical Information Circular IFR: Airborne Collision Avoidance System (ACAS) II Mandatory Carriage Requirements, Policy and special Exemptions*, AIC IFR 13/2003, Deutsche Flugsicherung, Langen, 2003, Download: [http://www.icao.int/anb/panels/scrsp/Meetings/SCRSP1/YELLOW%20COVER%20REPORT/A.I.1/WP-24-%20ATT-3%20-%20AIC%20IFR%2013%20\(03\).pdf](http://www.icao.int/anb/panels/scrsp/Meetings/SCRSP1/YELLOW%20COVER%20REPORT/A.I.1/WP-24-%20ATT-3%20-%20AIC%20IFR%2013%20(03).pdf)
- [ASX CAT21] EUROCONTROL: *Surveillance Data Exchange Part 12: Category 021 – ADS-B Reports*. Index: SUR.ET1.ST05.2000-STD-12-01, Brüssel 2009, Download: <http://www.eurocontrol.int/asterix/gallery/content/public/documents/cat021p12ed13.pdf>
- [ATA67] ATA: *Airborne Collision Avoidance System – Statement of Airline Policy, Requirements and a Technical Description of the System*, ATA-CAS Working Group: Washington D.C. Air Transport Association : 1967
- [BFU04] BFU: *Untersuchungsbericht AX001-1-2/02 zum Unfall am 01.Juli 2007 nahe Überlingen/Bodensee*, Braunschweig: Bundesstelle für Flugunfalluntersuchung 2004, Download: http://www.bfu-web.de/cln_015/nn_223970/DE/Publikationen/Untersuchungsberichte/2002/Bericht_02_AX001-1-2
- [Bredemeyer03] Jochen Bredemeyer, Peter Form, *Experimentally tested detection of interfered Mode S replies*, Proceedings of International Radar Symposium (IRS 2003), Dresden
- [DFS04] DFS: *ATM-Systeme: Mode S in betrieblicher Nutzung*. Pressemitteilung der Deutschen Flugsicherung vom 02. April 2007, Langen, Download: http://www.dfs.de/dfs/internet_2008/module/presse/deutsch/presse/presseinformation/2007/atm_systeme_mode_s_in_betrieblicher_nutzung_2_4/index.html

- [DLR IfEV08] *Kooperationsvertrag auf dem Gebiet der Mode S-Systeme in der Luftfahrt zwischen dem Institut für Eisenbahnwesen und Verkehrssicherung der TU Braunschweig Abteilung Sicherung im Luftverkehr, el. Verkehrssicherung und Deutschen Zentrum für Luft und Raumfahrt e.V. (DLR) handelnd für sein Institut für Flugführung (DLR-FL), Braunschweig 2008*
- [DrummKuchar07] Ann C. Drumm, James K. Kuchar: *The Traffic Alert and Collision Avoidance System*. Lincoln Laboratory Journal Volume 16 Number 2, 2007, Download: http://www.ll.mit.edu/publications/journal/pdf/vol16_no2/16_2_04Kuchar.pdf
- [EurocontrolPASS] Eurocontrol: *PASS Project: Analysis of RA Downlink data*. Dokument-ID: PASS/WA1/WP4/38, Brüssel 2009, Download: http://www.eurocontrol.int/ra-downlink/gallery/content/public/library/PASS_WA1_WP4_38W_v1.5.pdf
- [FAA TSO-C119A] FAA: *Technical Standard Order Subject Traffic Alert and Collision Avoidance System (TCAS) Airborne Equipment, TCAS II*, TSO-C119A, Department of Transportation, Federal Aviation Administration, USA, Washington, D.C. 1990, Download: http://rgl.faa.gov/Regulatory_and_Guidance_Library/%5CrgTSO.nsf/0/2CF27F43101ABD1386256E930055ED80?OpenDocument
- [Form74] Peter Form: *Digitale Synchronisation für zeitsynchrone Kollisionsschutzsysteme in der Luftfahrt*, Habilitationsschrift, TU Braunschweig 1974, DGLR-PAPER-73-012
- [Form83] Peter Form: *Kollisionsschutz in der Luftfahrt – Eine Studie für den Bundesminister für Verkehr*, Institut für Verkehr, TU Braunschweig
- [Form86] Peter Form: *Considerations on the Rules of the Air and compatible Resolution of IFR/VFR Conflicts by TCAS II*, Arbeitsgruppe 2 des ICAO SICAS-Panel, WP2/104, TU Braunschweig, Canberra 1986
- [Form90] H. Rohling, P. Form, G. Schänzer, et. Al.: *Auswirkungen neuer Technologien auf die Sicherheit im Luftverkehr*, Deutsche Gesellschaft für Ortung und Navigation (DGON), Deutsche Gesellschaft für Luft- und Raumfahrt e.V. (DGLR), Symposium Sonderforschungsbereich 212, Braunschweig 1990, ISBN: 3-88585-870-3
- [Herber90] Andreas Herber: *Untersuchungen von Bordkollisionsschutzsystemen für die internationale Zivilluftfahrt*, Dissertation, Institut für Verkehr, TU Braunschweig 1990 ISBN: 3-9802073-6-6
- [ICAO Annex10v4] ICAO: *Annex 10 to the Convention on International Civil Aviation – Aeronautical Telecommunications – Volume IV – Surveillance Radar and Collision Avoidance Systems*. Montreal: International Civil Aviation Organization, Third Edition: 2002, ISBN 92-9194-952-3
- [ICAO98] ICAO: *Report on the fourth Meeting of SICASP*, Montreal: International Civil Aviation Organization 1989
- [ICAO91] ICAO: *Procedures for the International Operational Evaluation of ACAS II*, ICAO State Letter AN 7/11.12-91/11: Montreal International Civil Aviation Organization 1991

- [ICAO93] ICAO: *Draft Standards and Recommended Practices (SARPS) for Airborne Collision Avoidance Systems (ACAS)*, SICASP/5-DP/3, ICAO-SICAS-Panel-Meeting 5, Montreal: International Civil Aviation Organization 1993
- [ICAO9688] ICAO: *Manual on Mode S Specific Services*. Doc 9688, Montreal: International Civil Aviation Organization, Second Edition: 2004, ISBN 92-9194-407-6
- [ICAO9871] ICAO: *Technical Provisions for Mode S Services and Extended Squitter*. Doc 9871, Montreal: International Civil Aviation Organization, Second Edition: 2007, ISBN 978-92-9231-117-9
- [Jessen] Ingo Jessen: *Datenlink-Konzepte ATTAS/ATRA.*, DLR-IB 112-2006/43 Deutsches Zentrum für Luft- und Raumfahrt e.V. Braunschweig, 2006, DLR-Bibliothekseintrag: <http://elib.dlr.de/46619/>
- [Kügler95] Dirk Kügler: *Potential von Kollisionsschutzsystemen an Bord ziviler Flugzeuge, Dissertation*, Institut für Verkehr, Eisenbahnwesen und Verkehrssicherung der TU Braunschweig, 1995: ISBN 3-8265-0861-0
- [Mensen] Heinrich Mensen: *Moderne Flugsicherung: Organisation, Verfahren, Technik*. 3. Auflage, Springer Verlag, 2004, ISBN: 3-540-20581-0
- [Orlando01] Orlando, Vincent: *1090 MHz ADS-B TCAS RA Broadcast*. Working Paper 1090-WP-3-02, RTCA Special Committee 186 on ADS-B 1090 MOPS Revision A, Working Group 3, Meeting #3, 2001, Download: http://adsb.tc.faa.gov/WG3_Meetings/Meeting3/1090-WP-3-02.pdf
- [Pietsch09] Stephanie Pietsch: *Programmierung eines ACAS-Prototypen*. Studienarbeit am Institut für Eisenbahnwesen und Verkehrssicherung, TU Braunschweig, Braunschweig 2009
- [Proakis] John G. Proakis, *Digital Communications, 3rd Edition*. McGraw-Hill Book Co., 1995 ISBN 0-07-113814-5
- [Plumeyer86] Peter Plumeyer: *Untersuchung synchroner und asynchroner Mess- und Übertragungsverfahren für den Kollisionsschutz in der Luftfahrt*, Dissertation, Institut für Verkehr, TU Braunschweig 1986
- [RTCA DO185] RTCA: *Minimum Operational Performance Standards for Traffic Alert and Collision Avoidance System (TCAS) Airborne Equipment*. RTCA DO-185, Washington D.C.: Radio Technical Commission for Aeronautics 1983
- [RTCA DO185A] RTCA: *Minimum Operational Performance Standards for Traffic Alert and Collision Avoidance System II (TCAS II) Airborne Equipment*. RTCA DO-185A, Washington D.C.: Radio Technical Commission for Aeronautics 1997
- [RTCA DO185B] RTCA: *Minimum Operational Performance Standards for Traffic Alert and Collision Avoidance System II (TCAS II)*. RTCA DO-185B, Washington D.C.: Radio Technical Commission for Aeronautics 2008

- [RTCA DO260A] RTCA: *Minimum Operational Performance Standards for 1090 MHz Extended Squitter Automatic Dependent Surveillance – Broadcast (ADS-B) and Traffic Information Services – Broadcast (TIS-B)*. RTCA DO-260A, Washington D.C.: Radio Technical Commission for Aeronautics 2008
- [Schröder09] Dorothee Schröder: *Automatische Charakterisierung von ACAS- Eingriffen in den Luftverkehr* , Studienarbeit am Institut für Eisenbahnwesen und Verkehrssicherung, TU Braunschweig, Braunschweig 2009
- [Schroer84] Wolfgang Schroer: *Kollisionsverhütung im Luftverkehr – Eine Systemanalyse unter besonderer Berücksichtigung transpondergestützter Bordsysteme mit selektiver Adressierung*, Dissertation, Institut für Verkehr, TU Braunschweig 1984
- [Thales05] *AS 680 ADS-B Ground-Station and RCMS – Raw Data Message*. Thales ATM GmbH , Stuttgart 2005
- [Thales06] *AS 680 ADS-B Ground-Station and RCMS – Technical Manual*. Thales ATM GmbH, Ref.-No. 83140 56121 , Stuttgart 2006
- [WGS84] NIMA: Department of Defense World Geodetic System 1984; Technical Report, TR 8350.2, 3rd edition; National Imagery and Mapping Agency January 2000, Download: <http://earth-info.nga.mil/GandG/publications/tr8350.2/wgs84fin.pdf>

Literatur aus der Informatik

- [AWK] Programmiersprache GNU Awk : <http://www.gnu.org/software/gawk/gawk.html>
alternativ:
Arnold Robbins: *GAWK: Effective AWK Programming: A User's Guide for GNU Awk*
O'Reilly, 2001, ISBN 0-596-00070-7, Download:
<http://www.gnu.org/software/gawk/manual/>
- [Bzip2] Freies Komprimierungs-Programm und -Bibliothek Bzip2: <http://bzip.org>
- [Boost] Freie C++-Klassenbibliothek Boost: <http://www.boost.org/>
alternativ:
Björn Karlsson: *Beyond the C++ Standard Library. An Introduction to Boost*. Addison-Wesley, 2007, ISBN 0-321-13354-4
- [C++] Bruce Eckel: *Thinking in C++, Volume 1: Introduction to Standard C++*. 2. Auflage. Prentice Hall, 2000, ISBN 0-13-979809-9, Download: <http://jamesthornton.com/eckel/>
und
Bruce Eckel: *Thinking in C++, Volume 2: Practical Programming*. 1. Auflage. Prentice Hall, 2003, ISBN 0-13-035313-2, Download: <http://jamesthornton.com/eckel/>
- [Cormen01] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest: *Introduction to Algorithms*. The MIT Press 2001, ISBN 0-262-53196-8.
- [DLR F-LAN] Axel Winkler: *White Paper über technische Lösung zum Thema Netzanbindung „Mode-S Monitor“ im DLR Braunschweig* , T-Systems SfR GmbH, 2008

- [Debian] Linux Distribution Debian GNU/Linux: <http://www.debian.org/>
alternativ:
Frank Ronneburg: *Debian GNU/Linux 4 Anwenderhandbuch*. Addison-Wesley München 2007, ISBN 978-3-8273-2523-5, Download:
<http://debiananwenderhandbuch.de/>
- [Debian SA1571-1] Debian-Sicherheitsankündigung: *openssl – Voraussagbarer Zufallszahlengenerator*: DSA-1571-1, 13. Mai 2008, Download:
<http://www.debian.org/security/2008/dsa-1571>
alternativ:
cr (Pseudonym): *Schwache Krypto-Schlüssel in Debian, Ubuntu und Co.* Artikel in Zeitschrift c't Heft 12 2008, Heise Verlag, Hannover, kostenpflichtiger Download:
http://www.heise.de/kiosk/archiv/ct/2008/12/56_Sicherheit
- [GNUPlot] Diagramm-Beschreibungssprache GNUPlot: <http://www.gnuplot.info>
alternativ:
Philipp Janert: *Gnuplot in Action*. Manning, April 2009, ISBN 1-933-98839-8
- [Harris07] Robin Harris: *Why RAID 5 stops working in 2009*, Artikel auf zdnet.com 2007,
<http://blogs.zdnet.com/storage/?p=162>
- [Heise.de] Nachrichtenticker Heise Online: <http://heise.de> Heise Verlag, Hannover
- [Fedora] Linux Distribution Fedora Core: <http://fedoraproject.org/>
alternativ:
Chris Tyler: *Fedora Linux: A Complete Guide to Red Hat's Community Distribution*, O'Reilly Media, Inc.; 4 edition (April 1, 2004) ISBN: 0-596-00589-X
- [Knuth1] Donald Knuth. *Fundamental Algorithms*, 3. Ausgabe. Band 1 der Reihe *The Art of Computer Programming*, Addison-Wesley, 1997. ISBN 0-201-89683-4.
- [Knuth2] Donald E. Knuth: *Sorting and Searching*. 2. Ausgabe, Band 3 der Reihe *The Art of Computer Programming*. Addison Wesley Boston 2008, ISBN 0-201-89685-0
- [Linux] Open Source Betriebssystem Linux: <http://www.kernel.org>
alternativ:
Matt Welsh, Matthias Kalle Dalheimer, Terry Dawson, Lar Kaufman: *Linux. Wegweiser zur Installation & Konfiguration*. O'Reilly, Köln 2004, ISBN 3-89721-353-2,
Download: http://www.oreilly.de/german/freebooks/rlinux3ger/linux_wegIVZ.html
- [Python] Programmiersprache Python: <http://www.python.org>
alternativ:
Peter Kaiser, Johannes Ernesti: *Python – Das umfassende Handbuch – Aktuell zu Python 2.5*, Galileo Press, ISBN 978-3-8362-1110-9, Download:
<http://www.galileocomputing.de/openbook/python/>
- [Shannon] Claude Elwood Shannon und Warren Weaver: *The Mathematical Theory of Communication*, University of Illinois Press 1963, ISBN 0-252-72546-8

- [SSH] IETF: Standard: *The Secure Shell (SSH) Protocol*. RFC 4250 bis 4256, The Internet Engineering Task Force, 1980, <http://tools.ietf.org/html/rfc4250> ff.
alternativ:
Daniel J. Barrett, Richard Silverman, Robert Byrnes: *SSH, The Secure Shell: The Definitive Guide*, O'Reilly Media, Inc. Second Edition 2005, ISBN 978-0-596-00895-6
- [OSM] Freie Wiki-Weltkarte OpenStreetMap: <http://www.openstreetmap.org/>
alternativ:
Frederik Ramm, Jochen Topf: *OpenStreetMap – Die freie Weltkarte nutzen und mitgestalten*. Lehmanns Media, 2. erw. Auflage, Berlin 2009, ISBN 978-3-86541-320-8.
- [OOP] Bernhard Lahres, Gregor Raýman: *Praxisbuch Objektorientierung*. Galileo Computing, ISBN 3-89842-624-6, Download: <http://openbook.galileocomputing.de/oop/>
- [OpenSSH] Freies Softwarepaket zur verschlüsselten Kommunikation OpenSSH:
<http://openssh.com/>
alternativ:
Timo Dotzauer, Tobias Lütticke: *Das SSH-Buch. Leitfaden für den sicheren Einsatz von OpenSSH*. Lohmar, Millin, 2007, ISBN 3-938626-03-8
- [RAID] David A Patterson, Garth Gibson, Randy H. Katz: A Case for Redundant Arrays of Inexpensive Disks (RAID). Computer Science Division, Department of Electrical Engineering and Computer Science, University of California, Proceedings of the International Conference on Management of Data (SIGMOD), June 1988, Download: <http://www-2.cs.cmu.edu/~garth/RAIDpaper/Patterson88.pdf>
- [STL] GNU Standard C++ Library: <http://gcc.gnu.org/libstdc++/>
alternativ:
Nicolai M. Josuttis: *The C++ Standard Library – A Tutorial and Reference*, Addison-Wesley U.S.A., 1999, ISBN 0-201-37926-0
- [SVN] Frei Versionsverwaltung Subversion (SVN): <http://subversion.tigris.org/>
alternativ:
C. Michael Pilato, Ben Collins-Sussman, Brian W. Fitzpatrick: *Versionskontrolle mit Subversion*, O'Reilly-Verlag, 3. Auflage April 2009, ISBN 978-3-8972-1897-0, Download: <http://svnbook.red-bean.com/>
- [UDP] IETF: Standard: *User Datagram Protocol*. RFC 768, The Internet Engineering Task Force, 1980, <http://tools.ietf.org/html/rfc768>
alternativ:
Andrew Stuart Tanenbaum: *Computer Networks*, Prentice Hall, Upper Saddle River, NJ USA 1981, ISBN 0-13-066102-3
- [UML] Modellierungssprache Unified Modeling Language (UML): <http://www.uml.org/>
alternativ:
Grady Booch, James Rumbaugh, Ivar Jacobson: *Das UML-Benutzerhandbuch*. Addison-Wesley, 1999, ISBN 3-8273-1486-0

Eigene Beiträge


- [Gottst07] Jens Gottstein, Alexander Burkert, Prof. Dr.-Ing. Peter Form: *ACAS-Monitoring by an Experimental Groundstation in Braunschweig*, German Institute of Navigation DGON-ESAVS Konferenz 2007, Proceedings bestellbar unter <http://www.dgon.de>
- [Gottst08] Jens Gottstein, Prof. Dr.-Ing. Peter Form: *Langzeitbeobachtung von ACAS/TCAS Aktivitäten im Norddeutschen Luftraum*, Vortrag Deutsches-Flight-Safety-Forum Köln 13. März 2008
- [Gottst08b] Jens Gottstein, Prof. Dr.-Ing. Peter Form: *ACAS-Monitoring of 1 000 000 flight hours in the North German Airspace*, Enhanced Surveillance of Aircraft and Vehicles Symposium, 2008 Capri, Proceedings erschienen bei Centro Vito Volterra ISBN: 978-88-903482-0-4
- [Gottst09] Jens Gottstein, Prof. Dr.-Ing. Peter Form: *Two Million Flight Hours ACAS-Monitoring*, Proceedings of the 2nd International Workshop on Aircraft Systems Technologies, 2009 Hamburg, Proceedings erschienen bei Shaker Verlag Aachen, ISBN 978-3-8322-8071
- [Gottst09b] Jens Gottstein, Prof. Dr.-Ing. Peter Form: *Five Million flight hours continuous Reception of ACAS-Communications and reporting of ACAS/TCAS-interventions in the German Airspace*, EUROCONTROL Safety R&D Seminar, München, 21. bis 22. Oktober 2009, download:
[http://www.eurocontrol.int/eec/gallery/content/public/document/other/conference/2009/safety_r_and_d_Munich/day_2/Jens-Gottstein-\(TU-Braunschweig\)-Paper.pdf](http://www.eurocontrol.int/eec/gallery/content/public/document/other/conference/2009/safety_r_and_d_Munich/day_2/Jens-Gottstein-(TU-Braunschweig)-Paper.pdf)

Verwendete Abkürzungen und Einheiten

AA	Announced Address, Bezeichnung für Mode S-Adresse
ACAS	Airborne Collision Avoidance System nach ICAO Standard, unter ACAS wird meist ACAS II verstanden
ACAS I	ACAS Version I, basiert auf Mode A/C-Transponder. Kann sich nicht koordinieren und war nie Pflichtausrüstung.
ACAS II	Weltweit verwendete Version von ACAS, Pflichtausrüstung in Zivilluftfahrt, kann nur vertikale Ausweichenweisungen geben
ACAS III	Weiterentwicklung von ACAS II, die aufgrund verbesserter Messtechnik auch horizontale Anweisungen geben soll, wurde nie entwickelt.
ACAS IV	Angedachtes zukünftiges ACAS, das ADS-B-Daten auswerten soll und vertikale RA geben soll
ACAS-Event	Ein vom ACAS-Monitor erkanntes und dokumentiertes RA
ACAS_Events_DB	Datenstruktur im ACAS-Monitor, die alle gerade in der Auswertung bearbeiteten ACAS-Events speichert
ACASMON	ACAS-Monitor-Software, in dieser Arbeit entwickelt
ADS-B	Automatic Dependent Surveillance Broadcast, Standard, nach dem Flugzeuge ihre aktuelle Position und andere Daten selbstständig ohne Anfrage regelmäßig in DF17-Formaten aussenden
ADSL	Asymmetric Digital Subscriber Line, Anschlusstechnik von Breitbandanschlüssen für Konsumenten
Alt	Altitude, Höhe, wenn nicht anders angegeben Druckhöhe eines Flugzeuges
AP	Address Parity Feld in Mode S-Nachrichten, enthält gleichzeitig Prüfsumme und Absender- oder Empfängeradresse der Nachricht
ASTERIX	All Purpose Structured Eurocontrol Surveillance Information Exchange, Protokoll für den Austausch von Flugüberwachungsdaten
ASX	Abkürzung für ASTERIX
awk	Programmiersprache zur Auswertung einfacher Textdaten.
BFU	Bundesstelle für Flugunfalluntersuchung, Deutsche Untersuchungsstelle für Flugunfälle
bzw.	beziehungsweise
C++	objektorientierte Programmiersprache in der der ACAS-Monitor implementiert ist
ca.	circa
CAT	Category
COR	Corrective RA
CPR	Compact Position Report, Kodierung der Positionen im ADS-B
CRC	Cyclic redundancy check, zyklische Redundanzprüfung
DB	Datenbank
dB	Dezibel, eine Einheit für logarithmische Größen wie Spannungspegel usw.
dBm	Dezibel (Leistungspegel) mit der Bezugsgröße 1 mW
DF	Downlink Format im 1090 MHz Kanal des Mode S
DFS	Deutsche Flugsicherung
d. h.	das heißt
DL	Downlink, 1090 MHz Kanal im Mode S
DLR	Deutsches Zentrum für Luft- und Raumfahrt
DME	Distance Measuring Equipment, Entfernungsmessgerät Flugnavigation
DPSK	Differential Phase Shift Keying, Phasendifferenzmodulation

dt.	deutsch
EUROCONTROL	European Organisation for the Safety of Air Navigation, Organisation zur zentralen Koordination der Luftverkehrskontrolle in Europa.
FAA	Federal Aviation Administration, US-amerikanische Luftfahrtaufsichtsbehörde
FL	Flight Level, Flugfläche, Druckhöhe bezogen auf Normaldruck (1013,25 hPa) in 100 ft Inkrementen
FPGA	Field Programmable Gate Array, Integrierter Schaltkreis, in dem logische Schaltungen programmiert werden können
fpm	feet per minute, Fuß pro Minute, $1 \text{ fpm} = 1 \text{ ft/min} = 5.08 \cdot 10^{-3} \text{ m/s}$
ft	feet, Fuß, $1 \text{ ft} = 0,3048 \text{ m}$
ft/min	Fuß pro Minute, $1 \text{ ft/min} = 5.08 \cdot 10^{-3} \text{ m/s}$
GByte	Gigabyte = 10^9 Byte, Einheit der Datenmenge
GPS	Global Positioning System, satellitengestütztes System zur weltweiten Positionsbestimmung
GSM	Global System for Mobile Communications, Standard für volldigitale Mobilfunknetze
hex	Hexadezimal, Kennzeichnung von Zahlen zur Basis 16
hPa	Hektopascal, SI Einheit des Druckes
ICAO	International Civil Aviation Organization, Internationale Zivilluftfahrtorganisation der Vereinten Nationen
ICAO24	Mode S-Adresse
IfEV	Institut für Eisenbahnwesen und Verkehrssicherung der TU Braunschweig
IP	Internetprotokoll, Netzwerkprotokoll zur Vermittlung von Paketen im Internet
KByte	Kilobyte = 10^3 Byte, Einheit der Datenmenge
km	Kilometer
kts	Knoten $1 \text{ kts} = 1 \text{ NM/h} = 0.51 \text{ m/s}$
LAN	Local Area Network, ein lokales Netz in der Computertechnik
MByte	Megabyte = 10^6 Byte, Einheit der Datenmenge
MESZ	Mitteuropäische Sommerzeit, Differenz zur MEZ ist 1 Stunde, zur UTC 2 Stunden
MEZ	Mitteuropäische Zeit, Zeitzone, Differenz zur UTC ist 1 Stunde
MHz	Megahertz, Einheit für die Frequenz
min	Minute
Mode S	Sekundärradarsystem mit adressierter Abfrage und zugehöriges Übertragungsprotokoll
Mode A	Adressabfragemodus im klassischen nicht adressiert abfragenden Sekundärradar
Mode A/C	Klassisches nicht adressiert im Mode A und Mode C abfragendes Sekundärradarsystem
Mode C	Höhenabfragemodus im klassischen nicht adressiert abfragenden Sekundärradar
ms	Millisekunde
NAT	Network Address Translation, Verfahren, um automatisiert Adressinformationen in Datenpaketen zu ersetzen und um private Netze mit dem Internet zu verbinden
NM	Nautische Meile $1 \text{ NM} = 1852 \text{ m}$
ns	Nanosekunde
OOP	Objektorientierte Programmierung, Programmierparadigma das Daten und dazugehörige Funktionen zu Klassen zusammenfasst
PC	Personal Computer, Einzelplatzrechner
Plane_DB	Datenstruktur für das Umgebungsmodell im ACAS-Monitor, speichert Zustand aller Flugzeuge
POS	Positive RA
PPM	Pulsphasenmodulation
PREV	Preventive RA
RA	Resolution Advisory, Ausweichenweisung des ACAS

RAC	RA-Complement, Komplementäre Ausweichenweisung
RAID	Redundant array of independent disks, redundante Anordnung unabhängiger Festplatten
RAM	Random-Access Memory, Halbleiterspeicher, z. B. für den Arbeits- oder Hauptspeicher
RTCA	Radio Technical Commission for Aeronautics, Standardisierungsorganisation in den USA
s	Sekunde
SATT	Safety trough Transparency Projekt das die Entwicklung des ACAS-Monitors initiierte
SCP	Secure CoPy ist ein Protokoll zur verschlüsselten Übertragung von Dateien über SSH
SPU	Signal Processing Unit, Thales 1030 MHz bzw. 1030 MHz Empfangsgerät in Mode S-Bodenstation
s. u.	siehe unten
SSH	Secure Shell, Netzwerkprotokoll zum verschlüsselten Zugriff auf entfernten Computer
SSR	Secondary Surveillance Radar, Sekundärradar, Radar mit aktiven kooperativen Zielen
STL	Standard Template Library, Standardbibliothek für C++
SVN	Subversion, ein Versionsverwaltungssystem für Quelltexte
TA	Traffic Advisory, Verkehrswarnung vom ACAS
TByte	Terabyte = 10^{12} Byte, Einheit der Datenmenge
TCAS	Traffic Alert and Collision Avoidance System, Name der Produkte nach ACAS-Standard und Bezeichnung von ACAS in RTCA-Standards und bei FAA
TCP	Transmission Control Protocol, verbindungsorientiertes (d. h. mit Flusskontrolle) Transportprotokoll im Internet
TCP/IP	Transmission Control Protocol / Internet Protocol, Familie von Netzwerkprotokollen, die die Basis des Internets bilden
TMA	Terminal Area, auch Terminal Control Area, Flughafennahbereich
UDP	User Datagram Protocol, verbindungsloses (d. h. ohne Flusskontrolle, anders als bei TCP) Transportprotokoll im Internet
UF	Uplink-Format im 1030 MHz Kanal des Mode S
UL	Uplink, 1090 MHz Kanal im Mode S
UML	Unified Modeling Language, standardisierte Beschreibungssprache für Strukturen und Abläufe in objektorientierten Programmsystemen
UMTS	Universal Mobile Telecommunications System, Standard für Mobilfunknetze
USB	Universal Serial Bus, Bussystem für Computer
USV	Unterbrechungsfreie Stromversorgung für Computer
usw.	und so weiter
UTC	Universal Time Coordinated, die Koordinierte Weltzeit
VSL	Vertical Speed Limit RA
WAN	Wide Area Network, zu deutsch „großräumiges Netz“
WGS84	World Geodetic System 1984, geodätisches Referenzsystem
WLAN	Wireless Local Area Network, lokales Funknetz nach IEEE-802.11-Standard
z. B.	zum Beispiel



© Technische Universität Braunschweig
Campus Forschungsflughafen
Hermann-Blenk-Straße 27
38108 Braunschweig
Telefon +49 531 391-9822
Telefax +49 531 391-9808
info@campus-forschungsflughafen.de
www.campus-forschungsflughafen.de

ISBN 978-3-928628-56-3